1

Forecasting the Tornadic Intensities of Thunderstorms By Multivariate Techniques

by

Dale Robert Perry

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Department of Marine, Earth, and Atmospheric Sciences

Raleigh

1989

Approved by:

Allen J. Riordan

Charles E. Anderson
Chairman of Advisory Committee

90 02 21 101

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b. RESTRICTIVE MARKINGS
NONE

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE

3. DISTRIBUTION / AVAILABILITY OF REPORT
APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)

5. MONITORING ORGANIZATION REPORT NUMBER(S)
AFIT/CI/CIA-89-085

6a. NAME OF PERFORMING ORGANIZATION
AFIT STUDENT AT NORTH CAROLINA STATE UNIVERSITY

6b. OFFICE SYMBOL *(If applicable)*

7a. NAME OF MONITORING ORGANIZATION
AFIT/CIA

6c. ADDRESS *(City, State, and ZIP Code)*

7b. ADDRESS *(City, State, and ZIP Code)*
Wright-Patterson AFB OH 45433-6583

8a. NAME OF FUNDING / SPONSORING ORGANIZATION

8b. OFFICE SYMBOL *(If applicable)*

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS *(City, State, and ZIP Code)*

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.

11. TITLE *(Include Security Classification)* (UNCLASSIFIED)
Forecasting the Tornadic Intensities of Thunderstorms by Multivariate Techniques

12. PERSONAL AUTHOR(S)
Dale R. Perry

13a. TYPE OF REPORT
THESIS/~~DISSERTATION~~

13b. TIME COVERED
FROM ________ TO ______

14. DATE OF REPORT *(Year, Month, Day)*
1989

15. PAGE COUNT
145

16. SUPPLEMENTARY NOTATION
APPROVED FOR PUBLIC RELEASE IAW AFR 190-1
ERNEST A. HAYGOOD, 1st Lt, USAF
Executive Officer, Civilian Institution Programs

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL
ERNEST A. HAYGOOD, 1st Lt, USAF

22b. TELEPHONE *(Include Area Code)*
(513) 255-2259

22c. OFFICE SYMBOL
AFIT/CI

AFIT/CI "OVERPRINT"

Abstract

PERRY, DALE R. Forecasting the Tornadic Intensities of Thunderstorms by Multivariate Techniques. (Under the direction of Charles E. Anderson.)

An investigation on the feasibility of statistically characterizing tornadic and non-tornadic thunderstorms after cells develop is the basis of this research. Nine tornado outbreak cases during the years 1984 through 1988 were studied. The intensity of each tornado-producing thunderstorm cell was obtained from *Storm Data* for the particular outbreak case studied. Satellite as well as radiosonde data were used to determine the meteorological parameters describing thunderstorm behavior.

One of these parameters was the downstream mass flux of an anvil outflow plume (called UMAX) and the other was the rightward deviation angle of the anvil from the storm-relative flow (called MDA). These parameters were used in a two-dimensional kinematic anvil plume model implemented on an interactive computer data system which simulated the actual anvil plume as seen in satellite imagery.

To gain an aspect of the pre-storm environment, proximity soundings were used to compute low-level vertical wind shear and potential buoyant energy (PBE) for each outbreak case. These two parameters were used to further refine the relationship between UMAX and MDA to the storm environment. A modified Fujita scale rating system was employed to characterize each outbreak cell as a weighted mean called F_w and a weighted mean square called F_w^2. F_w was a better indicator of tornadic intensity than F_w^2.

next page w 54

UMAX and MDA were used as predictors in a bivariate regression model (two-variables) to predict the tornadic intensity of a given observed cell. Linear regression yielded promising results in correlating the data, and described a "rising-ridge" surface that related UMAX and MDA to both F_w. Attempts to relate PBE and shear to F_w yielded mixed results, and only accounted for a small fraction of the error variance in the data.

When all nine outbreaks were combined into one group of data, the coefficient of determination fell drastically, implying that that these rising-ridge surfaces did not conform to each other when treated as a group. Non-dimensionalizing the combined data by outbreak breakpoint number (the UMAX and MDA value that separates tornadic from non-tornadic cells) gave slightly better results.

To increase the amount of variance accounted for in the regression model, PBE and vertical wind shear were added as two additional predictors in a quadvariate (four-variable) model. Linear regression gave improved coefficients of determination in each of the nine cases, suggesting the use of variables characterizing the pre-storm environment improves the prediction of the behavior of thunderstorms.

However when the nine outbreaks were combined and then regressed with all four variables, only a slight improvement in the variance ensued. Non-dimensionalization of the data improved the coefficient of determination slightly.

An effort to stratify the entire data set, based on each outbreak's breakpoint value, was attempted. The distance of each case's breakpoint value from the origin was determined. This measure gave a basis of grouping outbreak cases based on this stratification scheme.

Bivariate and quadvariate regression was run on the resulting four groups of stratified data. Linear regression analysis of these individual groups gave coefficients of determination in the range of 0.58 to 0.76 when UMAX and MDA were used in the bivariate model, and in the range of 0.66 to 0.80 when PBE and shear were added to make a quadvariate linear regression analysis.

Quadratic regression analysis gave coefficients of determination in the range of 0.78 to 0.86 for the bivariate model, and in the 0.91 to 0.95 range for the quadvariate model. However, it is questionable whether the quadratic results can be used in the stratified scheme due to the possibility of over-fitting the model to the data.

Future research should be done to relate the stratified groups of data to other meteorological phenomena. These unknown parameters may explain the unique triggering mechanisms that occur from case to case. If this is successful, then one may be able to implement an operational forecast to predict a thunderstorm cell's tornadic potential.

Acknowledgements

I would like to express my sincere appreciation to the individuals who contributed time and effort in the completion of this thesis. I'd like to thank Professor Charles E. Anderson for his guidance, patience and editorial comments that made this thesis readable. I wish to also thank Professor John F. Monahan for putting up with my many last minute visits that made the statistics in this research presentable. Also I'd like to thank Professor Allen J. Riordan for being on my committee.

I owe thanks to Kevin J. Schrab for the use of his UMAX and MDA data as well as for his helpful suggestions. In this respect, a big thanks goes to Erik Rasmussen, a fellow OU alumnus and storm chaser, who provided me a working copy of his PBE program.

However, I wouldn't have been able to do this without the loving support and encouragement of my wife, Lorraine, and my son, Geoffrey. They were there when I needed them, and were understanding when I couldn't be there for all the little things. I also have to thank my parents, Bob and Jean, who were very supportive of me to further my education and gave me the opportunity to do so. In this regard, I also thank my sister, Marcia, and my brothers, Lee and Keith, who were there when I needed them.

I'd also like to thank Joe and Nancy Miller, who were very supportive in the work I was doing at NC State. I owe a special thank you to the U.S. Air Force for allowing me the opportunity to attend graduate school. And finally to those I haven't mentioned, especially to my fellow Air Force comrades, who made my stay at NC State enjoyable and memorable.

This research was supported under NASA Contract NAS8-36547 and NOAA Contract 50-WCNE-8-06058.

Table of Contents

Page

1. INTRODUCTION

1.1 Tornado Outbreak Classification

The tornado is defined as a violently rotating column of air, pendant from a cumulonimbus cloud, and nearly always observable as a funnel cloud or tuba - (a cloud column or inverted cloud cone, pendant from a cloud base) (Glossary of Meteorology, 1959). The fact that tornadoes are a rare, violent and destructive atmospheric phenomena responsible for about 100 deaths and $200 million property damage annually (Davies-Jones, 1982) understandably ellicits much concern beyond the meteorological community. This is compounded with a typical damage area 2 km long and 50 m wide (extreme cases have damage path lengths exceeding 150 km and widths of 3 km). The majority of tornadoes are deemed weak and short-lived, although 3.2% of the total tornado occurrences have resulted in deaths of one or more persons (Galway, 1983).

Climatological studies have shown that the majority of the tornado deaths which occur in severe weather watch areas are caused by outbreaks of six or more tornadoes (Galway, 1975). In fact, Galway (1975) found that outbreaks of ten or more tornadoes accounted for 73% of the tornado deaths during the period 1952 through 1973. A question arises as to how an outbreak should be defined, since a tornado "outbreak" can mean many things to many people. This is particularly true in some states where tornado occurrences are very rare.

Pautz (1969) and Galway (1975) refined the concept of a tornado outbreak by introducing three outbreak categories: small (6-10 tornadoes), moderate (11-20 tornadoes) and large (greater than 20 tornadoes). For this research an outbreak is taken to be five or more tornadoes in a given synoptic situation to insure an adequate amount of thunderstorm cells will be examined.

Galway (1977) spatially defines three general types of outbreaks as local, progressive and line. A local outbreak was defined as an outbreak confined to a roughly circular envelope of approximately 10,000 n mi^2. Local outbreaks also tend to be the shortest in duration among the three types. This type of outbreak would most likely be caused by one or several cells. A progressive outbreak was defined as an outbreak that advances from west to east with time. Being somewhat long in length (the distance between the first and last tornado report is normally greater than 350 nautical miles) it has the longest life span of the three outbreak types. This outbreak type is more typically the result of a small number of cells. Lastly, a line outbreak was defined as one limited in eastward movement normally oriented along a north-south axis. Tornadoes that occurred in a line outbreak tend to be widely spaced in location but closely spaced in time. This type of outbreak is one in which a significantly larger number of cells would be involved. For purposes of this research these classifications are important in determining how many cells to pick for an outbreak case.

1.2 Tornadic Thunderstorms Characterized By Satellite Data

Previous work by Anderson (1979) investigated the use of geosynchronous satellite imagery to identify tornadic thunderstorms. Anderson examined the characteristics of storm tops and their related cirrus outflow patterns in five tornado outbreaks. His research yielded three characteristics of thunderstorm anvil behavior when viewed from the satellite prospective: a cirrus plume was strongly displaced to the right of the ambient wind, in some cases up to 90 degrees; the cirrus outflow had anticyclonic rotation; and the outflow plume contained spiral bands extending clockwise from the center of the storm.

Further studies by Schrab (1988) and Anderson and Schrab (1988) centered on the possibilility of identifying potential tornadic thunderstorms using characteristic signatures produced by the anvil emerging from a thunderstorm cell. These signatures were obtained by inputing parameters into a two-dimensional fluid simulation model implemented on the McIDAS (Man-computer Interactive Data Access System), a remote interactive computer terminal linked to the University of Wisconsin, Madison. These characteristic signatures were called UMAX and MDA. The parameter UMAX was defined as the maximum flux of radial outflow of anvil material defining a simulated cloud anvil plume. It has been shown that mass flux is directly related to tornadic intensity (Colquhoun and Shepherd, 1985). The other parameter, MDA (measured deviation angle), was defined as the clockwise deviation of the centerline of the anvil from the storm-relative ambient wind at the outflow level. Anderson (1979) suggested a connection between anvil outflow patterns and tornadic thunderstorms, which MDA tries to describe.

The anvil model simulates how an anvil outflow plume appears after time. The model does this by injecting an assemblage of particles into a two-dimensional fluid flow defined by a constant ambient wind plus a combination of Rankine vortex with or without compensation of vorticity farther out (Schrab, 1988). Thus the computed particle outline simulates the thunderstorm anvil envelope defined by UMAX and MDA. Schrab used this model in conjunction with available upper air data and a loop of three consecutive satellite pictures taken in a Lagrangian format at 1 km resolution in the visible and 4 km resolution in the IR. McIDAS allowed the user to isolate the representative temperature level of the anvil during this loop sequence, enabling the ambient wind at the anvil level to be determined. Care was also taken to avoid thunderstorm cold tops (overshooting the tropopause) and anvil edges, as these are not representative of the actual anvil plume being measured. By looping through the satellite images and overlaying the anvil envelopes, an assessment was made as to how well the simulated anvil matched the actual anvil plume. If the simulated envelope was too small then an increase in UMAX was needed. If too large, then a decrease in UMAX was input into the model. Also examined was the simulated anvil plume deviations, or measured deviation angle (MDA), from the storm relative flow. If the simulated anvil envelope was too far right of the actual anvil then a decrease in the clockwise tangential circulation parameter was input into the model. If too far left, then an increase in the parameter was needed. Thus, the speed of growth of the simulated anvil plume was controlled in order to match the observed satellite imagery.

1.2.1 Bivariate Regression of UMAX and MDA

A bivariate statistical model was chosen to relate UMAX and MDA as predictors to the response variable of tornadic intensity. This model is given by the following equations:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \varepsilon \qquad \{1.1\}$$

and,

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \beta_{12} x_1 x_2 + \varepsilon \qquad \{1.2\}$$

where $\hat{y}$ is the response variable for tornadic intensity and x_1 and x_2 are the predictor variables. Equation 1.1 is the linear model with three terms, and equation 1.2 is the quadratic model with six terms. Table 1.1 shows the individual outbreak results from this analysis.

Table 1.1. Percent of variation (coefficient of determination or R^2) in tornadic intensity (F_w^2) that can be explained by the predictor variables UMAX and MDA for linear and quadratic bivariate regression.

Case		Linear R^2	Quadratic R^2
OK	(84117)	0.3680	0.6855
WI-IL	(84118)	0.8031	0.9813
IA	(84159)	0.6412	0.9022
NE	(85130)	0.8270	0.9836
OH-PA-NY	(85150)	0.7267	0.8975
SD	(86209)	0.7268	0.9121
KS	(86261)	0.5531	0.7541
TX-LA	(87319)	0.7490	0.8958
NC-VA	(88333)	0.8254	0.9525

From the linear analysis, UMAX and MDA had a coefficient of determination of over 69% in predicting tornadic intensities of thunderstorms. Quadratic regression analysis of UMAX and MDA had an R^2 value of over 88% in predicting tornadic intensity. However, quadratic regression analysis cannot be applied to individual outbreak cases. This was due to the relatively small number of data observations per outbreak set, causing the quadratic model to "overfit" the data. This overfit gives a coefficient of determination higher than it actually would have under an ideal distribution. Using this bivariate model, a dataset would need approximately 40 observations to overcome this overfit problem.

Looking at the linear results of table 1.1 suggests for a particular outbreak investigated, there appeared to be a strong connection of thunderstorm anvil characteristics to the associated tornadic intensities. In fact, a "rising-ridge" (Box and Draper, 1987) statistical surface resulted relating UMAX and MDA to tornadic intensity. This surface showed that as one ascends the ridge higher values of tornadic intensities can also be expected, and defined a breakpoint value separating tornadic from non-tornadic occurrences.

When the data were grouped into one data set, the coefficient of determination fell to just over 55%. This showed that when all nine cases were combined into one data set, the strong correlation between the predictors and the response variable was not present. This result suggests that other influences may be present to characterize cells as tornadic or non-tornadic. Even attempts to non-dimensionalize the data based on individual outbreak breakpoint values showed little improvement in the correlations. Non-dimensionalizing was done by dividing each outbreak's cell UMAX and MDA by its representative breakpoint value. For example, if a given outbreak had a breakpoint value for UMAX as 16 and MDA as 20, then all the cell's UMAX values were non-dimensionalized by dividing by 16. Likewise all the cell's MDA values were divided by 20. When applied to the other outbreaks, this gave an overall breakpoint value of 1 and 1. In the linear analysis scheme, the R^2 value increased to 57%, an increase of only to 2%.

1.3 Tornadic Thunderstorms Characterized by Raob Data

Previous studies have related severe local storm development to vertical wind shear and available buoyant potential energy of the environment in which the storm develops through the use of proximity soundings and/or by numerical simulation. Work by Blechman (1979) showed that as the low-level vertical wind shear increased so did the vorticity. Vertical wind shear in the low-levels (0-4km) has also been suggested to be connected to the development of tornadic storms (Rasmussen and Wilhelmson, 1983). Further, a combination of this low-level shear in combination with low-level buoyancy has been shown to have an effect on vorticity production (Weisman and Klemp, 1982; Rasmussen and Wilhelmson, 1983). Both studies characterized tornado producing storms as having large values of shear and large amounts of buoyant energy. An attempt to relate the two as an operational index (Leftwich and Wu, 1987) yielded mixed results when trying to relate this type of index to indicate violent tornado potential.

In addition to UMAX and MDA, potential buoyant energy (PBE) and low-level vertical wind shear were added to the bivariate model to make a quadvariate model (four variable). This showed some promise when related to the tornadic intensity of individual thunderstorm cells. This is due to the fact that tornadic thunderstorms form in environments characterized by large potential instability and large vertical wind shear. The latter can be computed fairly easily with available sounding data.

1.4 Research Goals/Objectives.

The basic goal of this research was to eliminate the two-step process that Schrab (1988) and Anderson and Schrab (1988) described in relating UMAX and MDA in a bivariate regression model to predict tornadic intensities of thunderstorms. That is, thunderstorm anvil behavior parameterized by UMAX and MDA showed a relationship on a plotted surface that distinquished tornadic from non-tornadic cells. We had hoped that the addition of the vertical wind shear in the low-levels (0-4 km) and PBE to the bivariate model, to create a four variable polynomial regression scheme, would be sufficient to predict the tornadic intensity of a given observed thunderstorm cell, regardless of its outbreak family.

Consequently, one objective of this research was to arrive at an improved prediction of the expected intensity of a given cell whose UMAX and MDA are predetermined by measurement using satellite data and by measuring the pre-storm environment (PBE and wind shear) computed from available proximity upper air soundings. This was done by the use of a quadvariate (four variable) regression model. When the model was applied to the nine outbreak cases, a fine-tuning only of the analysis results occurred due to the addition of PBE and shear to the model.

This led to the other objective of this research, that is, to combine the data in such a way as to improve the coefficient of determination. This was done by stratifying the lumped data set into different groups based upon how each individual outbreak's surface fell into two-dimensional space. By computing the distance of each outbreak's breakpoint value from the origin helped achieve this objective. This was done with the hope that the regression analysis results would improve, and thereby describe some commonality present in the individual outbreak data.

If the stratification scheme were successful, then we could move on to some sort of forecast in predicting the small fraction of tornadoes that turn out to be the killer tornadoes. Furthermore, an operational forecast conceivably could be implemented in which these parameters are automatically computed using real-time satellite and raob data in order to provide a more refined and accurate forecast of tornado occurrence in a tornado outbreak situation.

2. METHODOLOGY OF NINE TORNADIC OUTBREAK CASE STUDIES

2.1 Outbreak Situations Studied

This research studied nine tornado outbreak cases as listed in table 2.1. All cases involved tornadic outbreaks of five or more tornadoes with areal coverage encompassing more than one state. Appendix 6.1 gives a complete listing of individual cell distribution and statistics per outbreak.

Table 2.1. List of Tornado Outbreaks Used in This Research.

		No. Tornadoes/		
Outbreak		No. Cells/	No. Killed/	Damage Path
Region	Date	No. Torn. Cells	No. Injured	(in miles)
OK	26 Apr 84	7/16/3	0/5	106.0
WI-IL	27 Apr 84	7/16/5	3/38	46.7
IA	7 Jun 84	34/12/6	4/117	454.0
NE	10 May 85	8/13/4	0/4	148.0
OH-PA-NY	31 May 85	28/23/9	76/905	535.0
SD	28 Jul 86	14/12/3	0/1	77.5
KS	18 Sep 86	5/16/4	0/7	43.0
TX-LA	17 Nov 87	17/16/4	11/287	145.0
NC-VA	28 Nov 88	7/13/3	4/157	125.0
Totals		127/137/40	109/1682	

The Oklahoma outbreak of 26 April 1984 (figure 2.1) included cells in central Oklahoma into southeastern Kansas, and also central Iowa and South Dakota causing 5 injuries along damage tracks totaling 106 miles associated with 7 tornadoes. Sixteen cells were included in the population with three being tornadic.

The Wisconsin and Illinois outbreak of 27 April 1984 (figure 2.1) was associated with 7 tornadoes which killed 3 and injured 38, with two tornadoes reaching F4 in strength. This outbreak gave 16 cells to the population study, 5 of which were tornadic.

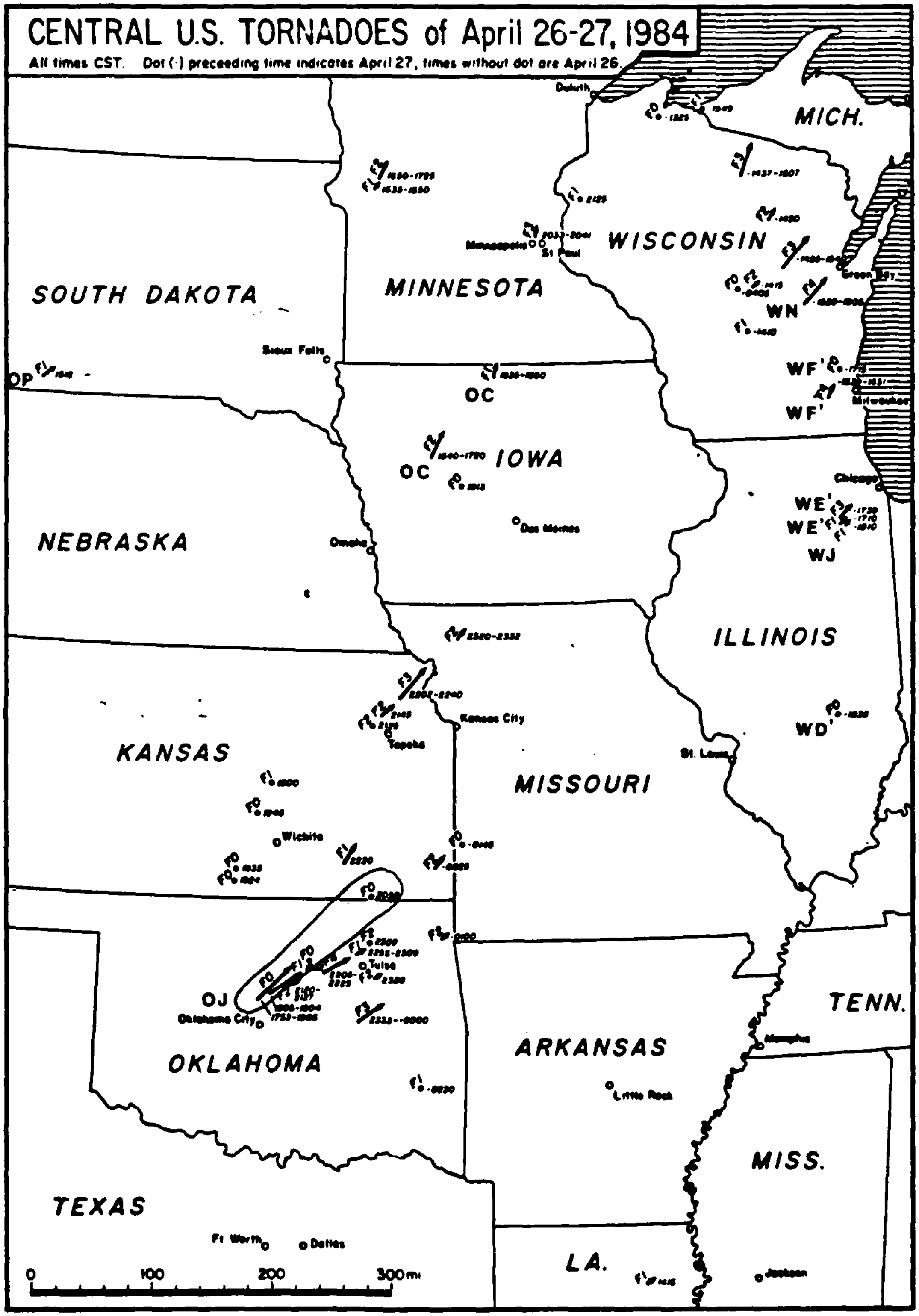


Figure 2.1. Tornado tracks and their associated cells for the OK and WI-IL outbreaks. Letters indicate the cell of the tornado's origin. Letters preceded by an O indicate cells from the OK outbreak; a W indicates cells from the WI-IL outbreak (after Storm Data).

The Iowa outbreak of 7 June 1984 (figure 2.2) produced 42 tornadoes encompassing extreme northeastern Kansas to southern Minnesota and central Wisconsin. Many of the tornadoes occurred in two clusters, one reaching from northeast Iowa into south-central Minnesota and north-central Wisconsin and the second extending from northeast Kansas and southeast Nebraska through northwest Missouri into south-central Iowa and then into south-central Wisconsin. The resulting deaths from these tornadoes were 13, with 319 injuries covering 584 miles of damage paths. In keeping with the limited scope of this research, only 34 tornadoes from the outbreak were used which contributed 4 deaths and 117 injuries to the dataset.

The 10 May 1985 Nebraska outbreak (figure 2.3) encompassed north-central Kansas to south-central Nebraska. This outbreak spawned 8 tornadoes along 158 miles of damage paths and resulted in only 4 injuries. One cell alone produced two F4 tornadoes in northern Kansas. This outbreak consisted of 13 observed cells, 4 of which were tornadic.

The Ohio, Pennsylvania, New York outbreak of 31 May 1985 (figure 2.4) produced the most devastating outbreak of tornadoes since the superoutbreak of 3-4 April 1974. Two separate outbreak areas were involved: one between Lake Ontario and Georgian Bay in Canada and the other centered in northwestern Pennsylvania stretching into eastern Ohio and southwestern New York. Of the 41 tornadoes produced, 28 were in the U.S. with 535 miles of damage paths and 13 were in Canada. These tornadoes caused 76 deaths and 905 injuries, and well over $100 million in property damage in the U.S. alone. This outbreak contributed 23 cells, 9 of which were tornadic, to the study

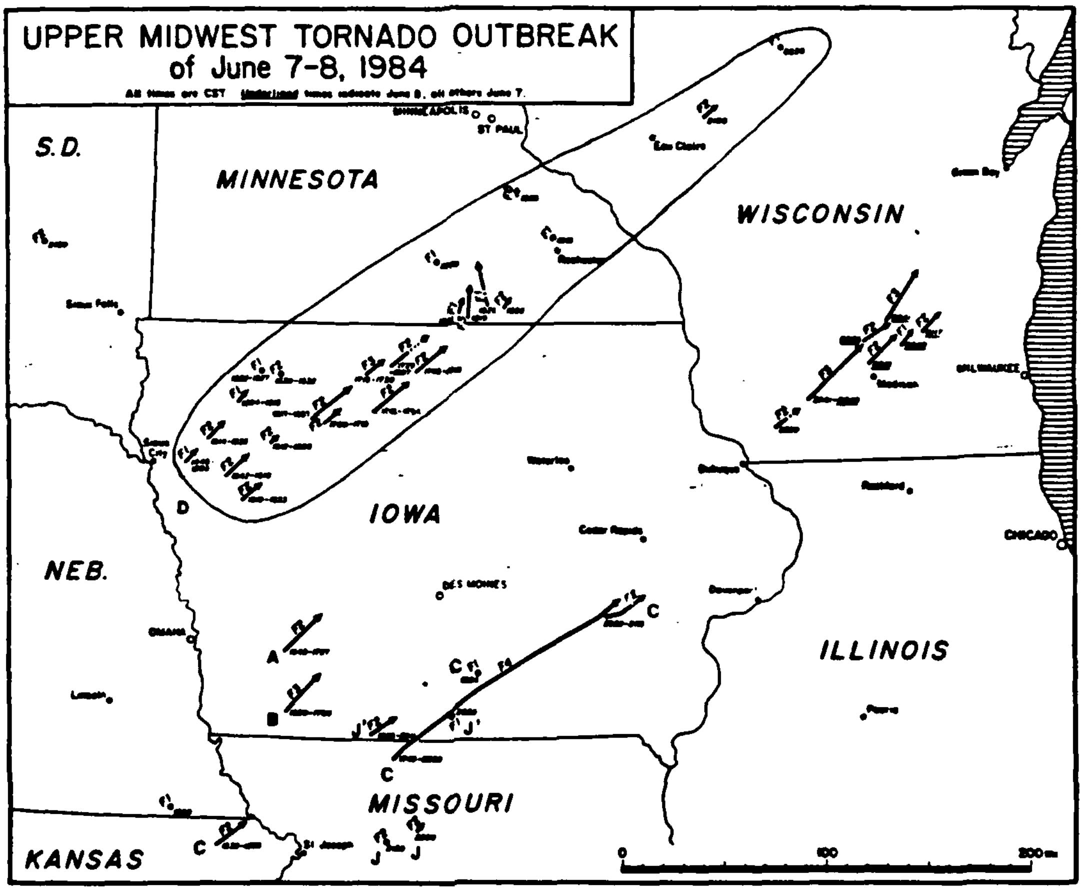


Figure 2.2. Tornado tracks and their associated cells for the IA outbreak. Letters indicate the cell of the tornado's origin (after Storm Data).

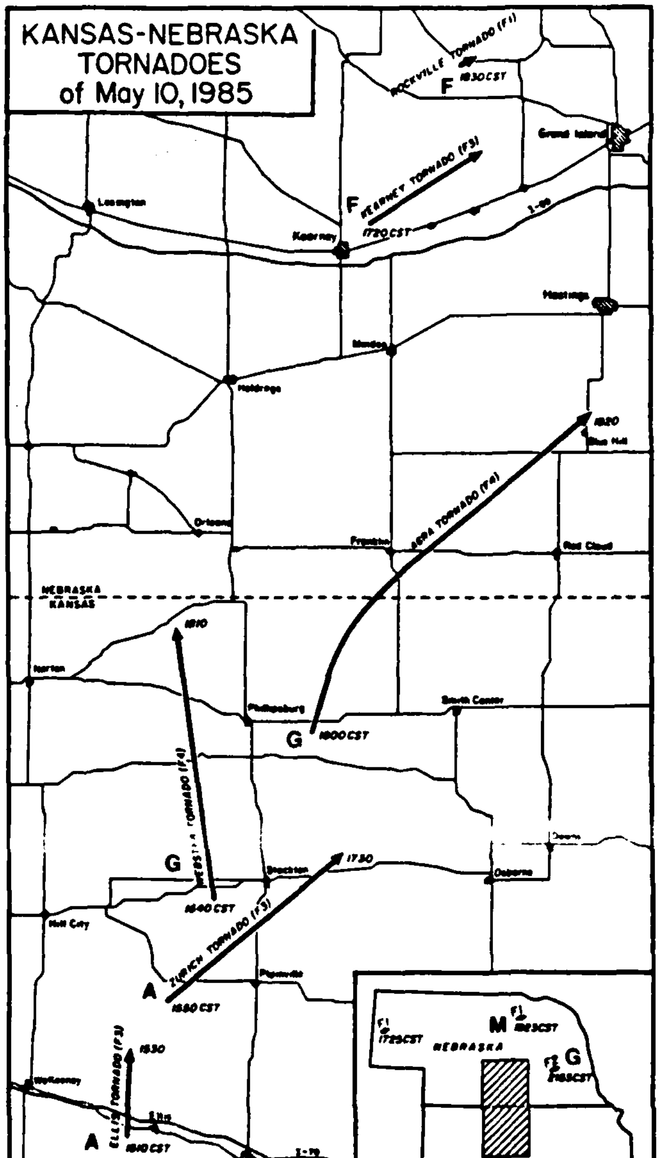


Figure 2.3. Tornado tracks and their associated cells for the NE outbreak. Letters indicate the.cell of the tornado's origin (after Storm Data).

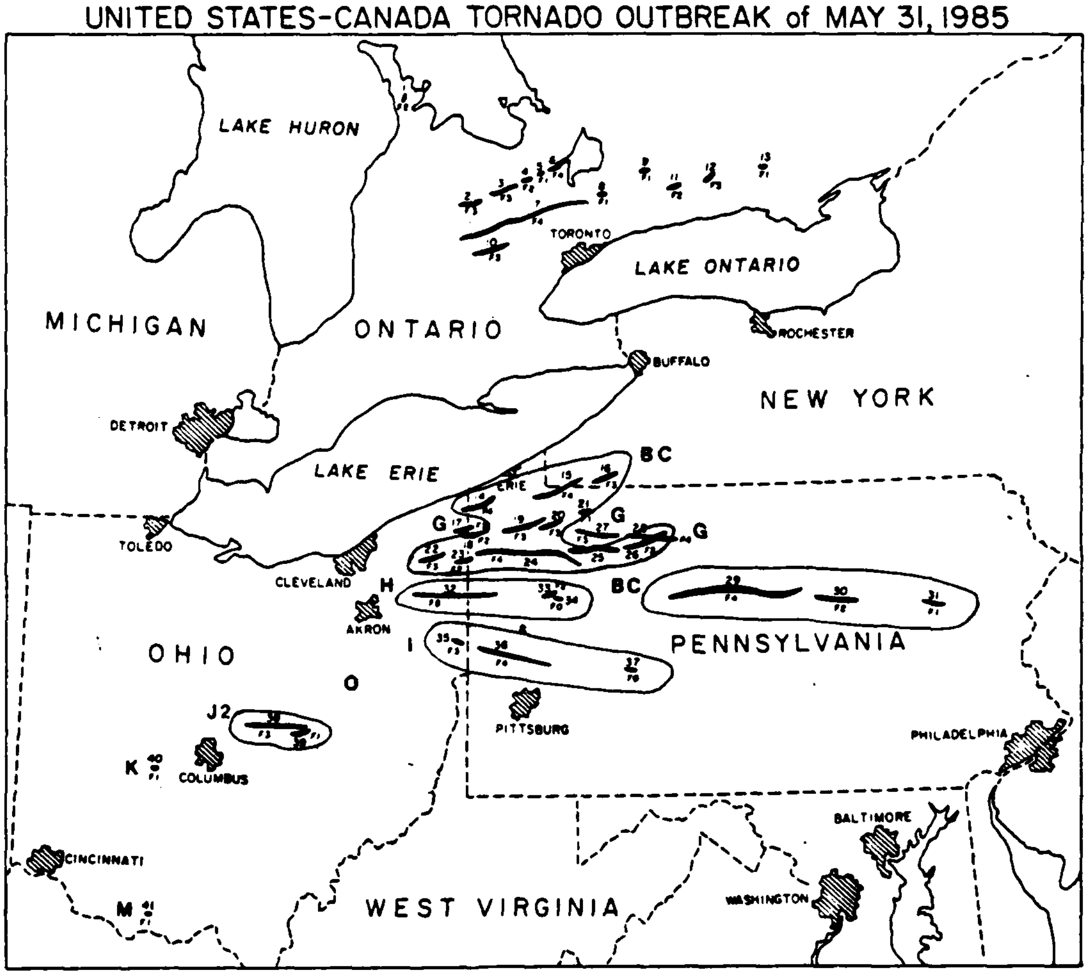


Figure 2.4. Tornado tracks and their associated cells for the OH-PA-NY outbreak. Letters indicate the cell of the tornado's origin (after Storm Data).

population.

During the afternoon of 28 July 1986, a group of thunderstorms formed in central South Dakota and moved southwest into a mesoscale convective complex (MCC) which moved subsequently across Iowa that evening. Fourteen tornadoes were produced over 78 miles of damage paths, one being F4 in intensity, with only one injury. This outbreak contributed 12 cells to the study population, 3 of which were tornadic.

The 18 September 1986 Kansas outbreak, stretching from northeastern Colorado into northern Kansas and south-central Nebraska, produced five tornadoes. Only seven people were injured along 43 miles of damage paths. This outbreak contributed 16 cells to the study population, of which only 4 were tornadic.

The Texas and Louisiana outbreak of 15 November 1987 (figure 2.5) occurred in eastern Texas and moved to extreme northwestern Louisiana, producing 17 tornadoes that killed 11 persons and injured 287.

The North Carolina and Virginia outbreak of 28 November 1988 (figure 2.6) covered two areas: one from Raleigh in central North Carolina to south-central Virginia, while the other occurred along the coastal areas of North Carolina. There were 7 tornadoes produced with 4 deaths and 157 injuries along 125 miles of damage paths. This outbreak contributed 13 cells to the study population, 3 of which were tornadic.

2.2 Classification of Tornadic Outbreak Intensities

In order to achieve the goals and objectives set out by this research, a tornadic strength scale (Schrab, 1988) was utilized to classify each thunderstorm cell examined. Storm Data was used to match

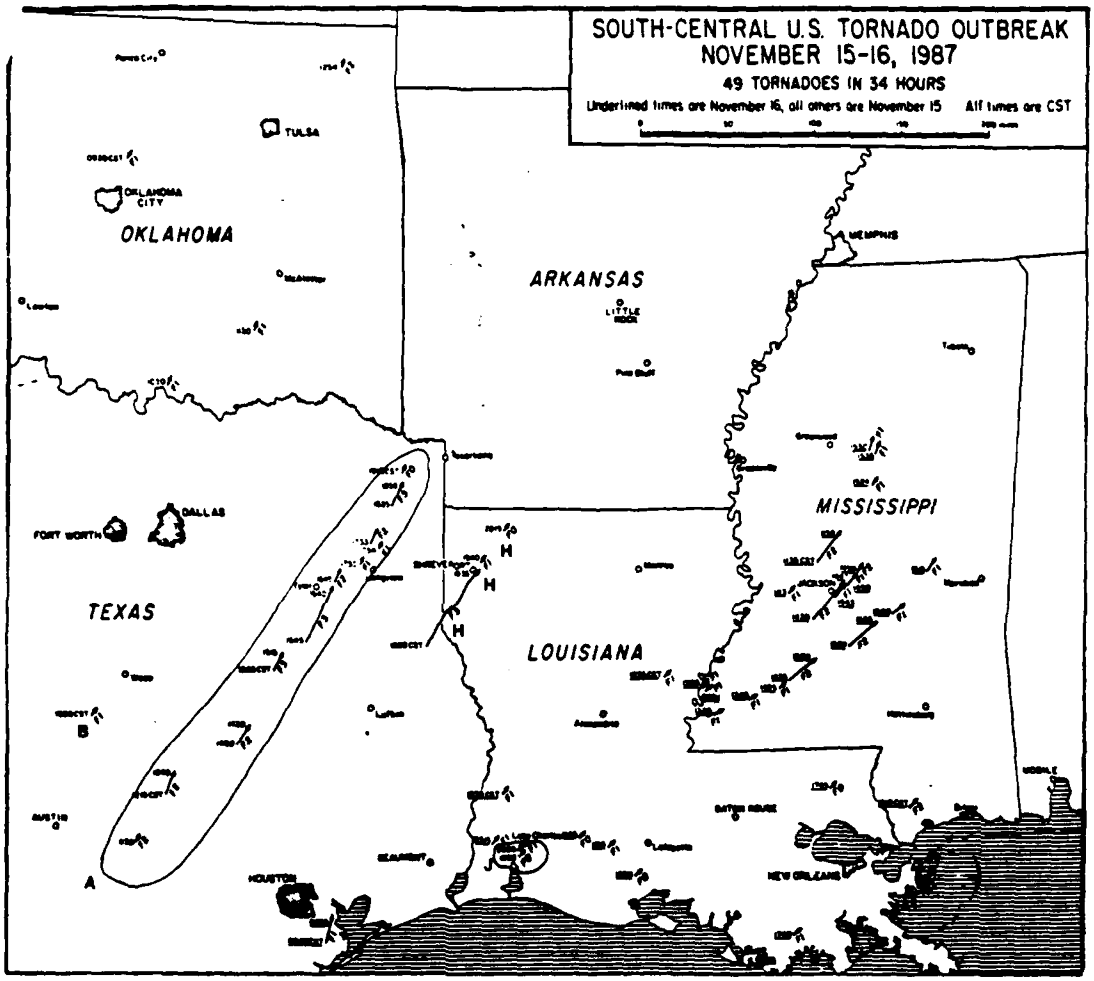
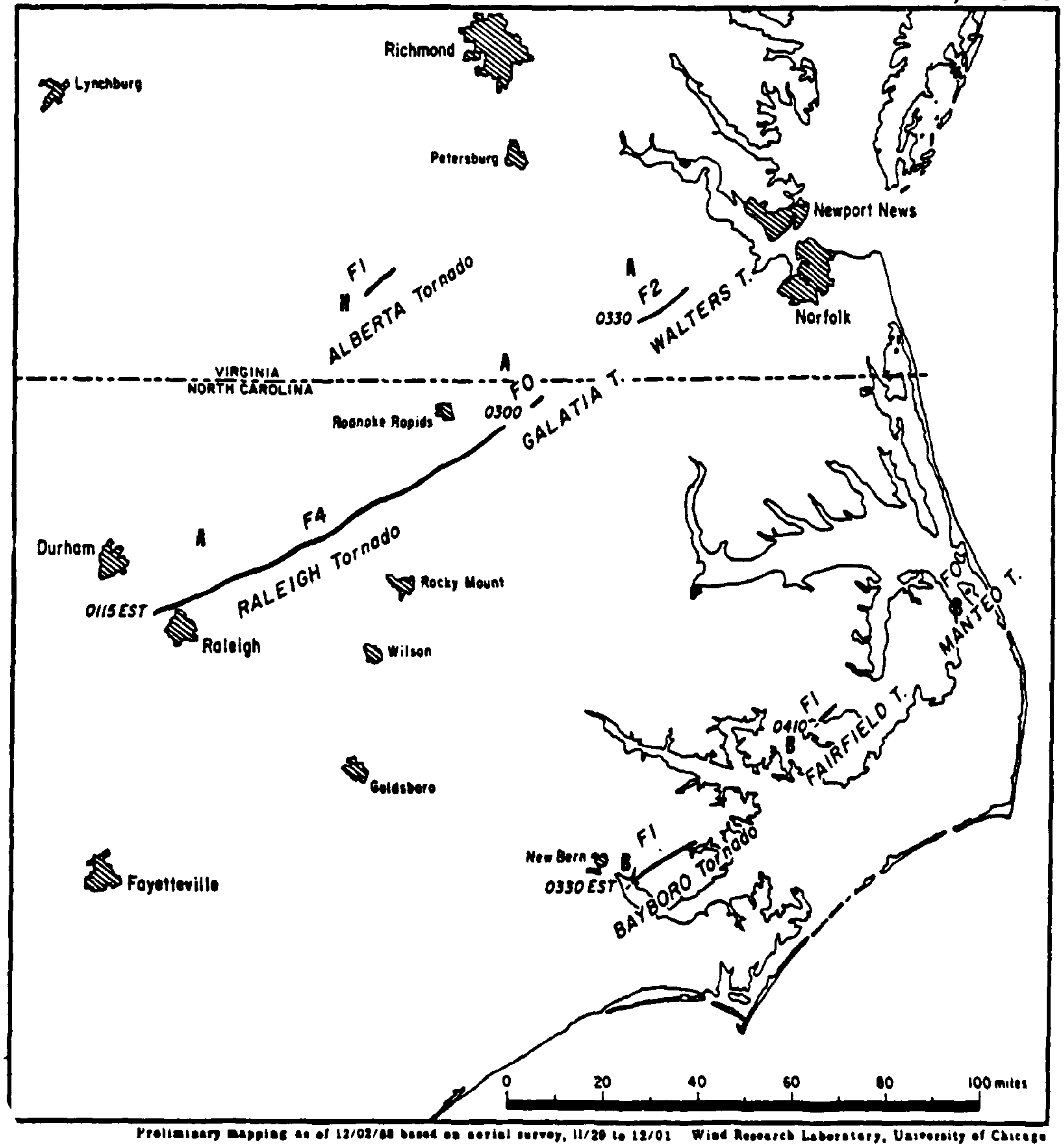


Figure 2.5. Tornado tracks and their associated cells for the TX-LA outbreak. Letters indicate the cell of the tornado's origin (after Storm Data).

NORTH CAROLINA-VIRGINIA TORNADOES on November 27, 1988



Preliminary mapping as of 12/02/88 based on aerial survey, 11/29 to 12/01 Wind Research Laboratory, University of Chicago

Figure 2.6. Tornado tracks and their associated cells for the NC-VA outbreak. Letters indicate the cell of the tornado's origin (after Fujita, Univ. of Chicago).

an observed cell's time and location to a documented occurrence of severe weather. If the reported severe weather associated with the thunderstorm cell was a tornado, then the cell was assigned a value based on the Fujita scale rating system (Fujita, 1981). In many cases more than one tornado was attributed to a single cell, in which case a weighted average of tornadic intensity was taken. Appendix 6.2 contains an excerpt from Fujita's study with an explanation of his rating scheme. Table 2.2 shows the weighted intensities of each tornado category. Since the Fujita scale is based on a natural logarithmic scale, the F_w^2 weightings in this table also shows this logarithmic relationship. A value of one was added to each F-scale category so that a log of F0 could be taken, shown as F_w in table 2.2. These values were then squared to get the weighted-squared Fujita scale, denoted as F_w^2. Figure 2.7 shows that the results fell on a straight line when the natural logarithm of each value was taken. The x-axis shows $\ln(F_w^2)$, the weighted squared Fujita scale. The y-axis shows $\ln(F_w)$, the weighted modified Fujita scale. Note that the F5 F_w value in table 2.2 was increased from 6.00 to 6.50 (36.00 to 42.25 for

Table 2.2. Modified Fujita Weighting Scale (after Schrab, 1988).

Fujita Scale Rating	Modified Tornado Intensity Scale	
	F_w	F_w^2
F0	1.00	1.00
F1	2.00	4.00
F2	3.00	9.00
F3	4.00	16.00
F4	5.00	25.00
F5	6.50	42.25

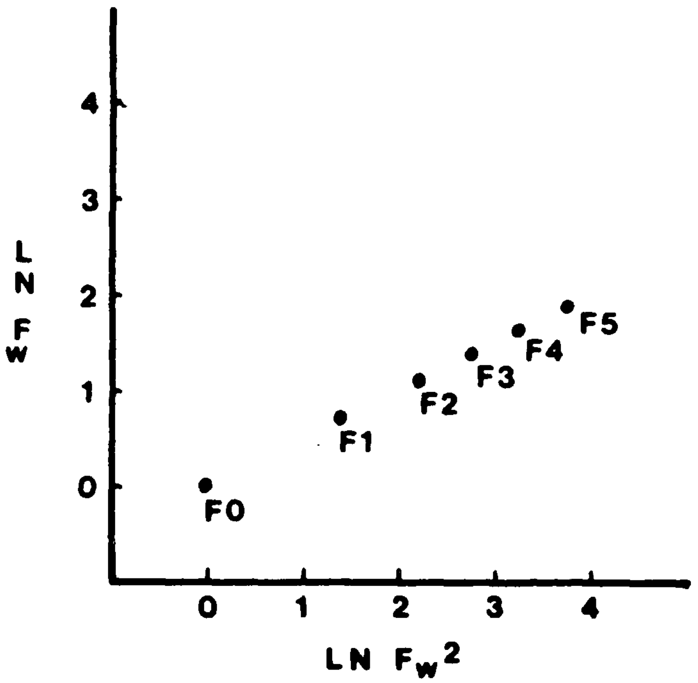


Figure 2.7. Natural log of modified weighted Fujita scale (F_W) versus natural log of weighted-squared Fujita scale ($F_W{}^2$) (after Schrab, 1988).

$F_w{}^2$) so that the distance between F4 and F5 shown in figure 2.7 was equal to the distance between F3 and F4 along both the x- and y-axes.

In order to account for the tornadic productivity of a given thunderstorm cell in an observed outbreak, a weighted mean square of all tornadoes connected with that particular cell was determined. For example, assume a cell spawned the following tornadoes: (1)F0 (2)F1 (1)F2 (3)F3 and (2)F4. This cell would then be assigned a weighted mean of 2.78 based upon the following calculation:

$$\frac{1(1) + 2(2) + 1(3) + 3(4) + 2(5)}{\text{Total Number of Tornadoes (9)}} = 2.78 \qquad \{2.1\}$$

Similarly, the weighted squared mean of this cell would compute to a value of 12.89 based upon the following calculation:

$$\frac{1(1) + 2(4) + 1(9) + 3(16) + 2(25)}{\text{Total Number of Tornadoes (9)}} = 12.89 \qquad \{2.2\}$$

If a cell produced no tornadoes then a value of zero was assigned. This rating system was employed in all nine outbreak cases studied.

As the Fujita rating scale is subjective, it was difficult in some cases to rate a tornado's damage based on Storm Data. Since the Fujita rating system is the accepted way of classifying a tornado's strength, the inherent bias in this type of rating system is unavoidable.

2.3 Choice of Predictor Variables

The two parameters chosen to describe thunderstorm anvil behavior were UMAX and MDA, i.e. the anvil outflow strength and the measured storm relative deviation angle. These were estimated on McIDAS, by fitting to each thunderstorm cell observed by satellite an envelope enclosing the anvil outflow boundary as computed via a two-dimensional numerical model (Schrab, 1988).

Two other parameters, closely related to UMAX and MDA, were chosen to predict a thunderstorm's tornadic behavior: vertical wind shear and parcel buoyancy. Low-level vertical wind shear has been attributed to veering and increasing winds with height (Davies-Jones, 1983), an important effect in the role of mesocyclones in thunderstorms. Available buoyant potential energy, or instability, has also been found to be particularly important in influencing storm evolution and structure (Weisman and Klemp, 1982). For purposes of this research, this instability will be referred to as potential buoyant energy (PBE).

Further, the combination of low-level (0-4 km) vertical wind shear and parcel buoyancy has been shown to have a strong relationship to convective storm structure and evolution (Weisman and Klemp, 1982; Rasmussen and Wilhelmson, 1983).

To determine these latter two parameters, available proximity soundings to each observed cell were used, provided they were representative of the air mass in which the storm developed. Using McIDAS, the cell's latitude/longitude coordinates were obtained from the satellite imagery and plotted on an upper air station locator map. As an example, if the observed cell developed at 2100Z, the closest 1200Z upper air sounding was used to define the pre-storm environment of wind shear and PBE. Care was taken to choose a sounding that best described the air mass that defined that cell. However, many cells did not develop near any one station, so an average of two or more soundings near and/or upstream of the cell were taken. For purposes of this research, if a cell was within 50 nm of a sounding station it was assumed representative of that cell's environment. Also, if a cell developed soon after 0000Z, say 0100Z, the 1200Z sounding was used under the premise that the 0000Z sounding would not be available in "real-time" for a forecast to be made as to its tornadic potential (even though the 0000Z sounding may have been more representative than the 1200Z sounding).

This research did not account for any modifications to the sounding's profile during the course of time between balloon launches. That is, any type of air mass modification that may occur that could alter the profile of an available sounding was not considered. This method was used even though air mass modification did occur during some of the case studies examined, and in some cases, had a profound effect when relating PBE and wind shear to tornadic intensities. This should be looked at for future research.

2.3.1 Determination of Low-Level Vertical Wind Shear

Rasmussen and Wilhelmson (1983) defined the low-level vertical wind shear as the mean shear in the lowest 4 km above ground level. This was expressed as:

$$\text{MEAN SHEAR} = \frac{\displaystyle\int_0^{4\,\mathrm{km}} \frac{\partial \vec{V}}{\partial z}\,dz}{\displaystyle\int_0^{4\,\mathrm{km}} dz} \qquad \{2.3\}$$

The mean shear in equation 2.3 is approximated by computing the magnitude of the shear vector from each 200 m layer averaged over the lowest 4000 m. This method of integrating the shear vector magnitudes every 200 m over the 4 km layer and dividing by 4 km was followed in order to account for the possible existence of strong speed shear and turning (called looping) in the layer. The presence of looping in the lower layer indicates the existence of a low-level jet with winds backing somewhat and decreasing markedly in speed just above the jet (Rasmussen and Wilhelmson, 1983). Computing the mean shear as just a vector difference between the 0 and 4 km levels would tend to mask the presence of this looping. This would be important since large low-level shears are typically found in the soundings from tornado cases caused by the presence of large loop structures in the profile

(Rasmussen and Wilhelmson, 1983).

As an example, table 2.3 lists a sounding profile. Suppose shear was computed as a straight vector difference between the 0 and 4 km levels. This would give an estimate of the mean low-level vertical shear as 2.8188 x 10^{-3} sec^{-1}. Whereas the mean low-level shear computed as an integrated sum over the entire 4 km layer would give the estimate as 7.9842 x 10^{-3} sec^{-1}. Clearly, the large difference between these two values indicates that the looping present in this data profile is masked with the first method. Obviously soundings where the lowest 4 km wind data was missing could not be used in this research. Appendix 6.3 contains an explanation of and the source code for the program that processed the sounding data.

Table 2.3 Typical Upper Air Sounding Profile

P	Wind Dir	Wind Spd	Height
(mb)	(deg)	(m/s)	(m)
1015	180	5.0	0.0
1000	185	9.0	15.0
850	195	25.0	1500.0
700	205	20.0	3000.0
600	215	15.0	4000.0
500	230	24.0	5800.0
400	230	30.0	7500.0
300	235	32.0	9550.0
100	250	26.0	16400.0

2.3.2 Determination of Potential Buoyant Energy (PBE)

PBE was computed, like shear, from available proximity soundings to developing cells. This sounding data was interpolated to 200 m levels and the PBE was computed from the following expression (Rasmussen and Wilhelmson, 1983):

$$\mathrm{PBE} = g\int_{\mathrm{LFC}}^{\mathrm{EL}} \frac{T_P - T_E}{T_E}\,dz \qquad \{2.4\}$$

where LFC is the level of free convection, EL is the equilibrium level, and the subscripts P and E are the parcel temperature and the environmental temperature respectively. The LFC is the height at which a parcel of air lifted dry-adiabatically from the surface until saturated and lifted saturation-adiabatically thereafter would first become warmer (less dense) than the environment (AWS, 1969). The parcel would then continue to rise freely above this level until it becomes colder (more dense) than the environment.

If the parcel is to be lifted then the LFC is the height where the moist adiabat taken from the LCL below intersects the sounding curve. If the parcel is heated then the LFC is the height where the moist adiabat taken from the CCL below intersects the sounding. The LCL is the height at which a parcel of air becomes saturated when lifted dry-adiabatically. The CCL is the point of intersection of a sounding with the saturation mixing ratio line corresponding to the average mixiing ratio in the surface layer. The program that computes PBE uses an average mixing ratio over a mixed layer of 50 mb. This is the height of the base of cumuloform clouds which are produced by thermal convection solely from surface heating. The EL is the height where the temperature of a buoyantly rising parcel again equals the environmental temperature, the height at where the moist adiabat from the LFC intersects the sounding above.

Computed from equation 2.4, PBE represents the maximum buoyant energy possible assuming the sounding profile does not change above the LFC. Hence, PBE is the positive area on a sounding, if one assumes that the convective temperature is reached (figure 2.8). Appendix 6.4 lists the program used to find PBE. The positive area of a sounding is the area between the moist adiabat and the sounding between the LFC and the EL. This area is proportional to the amount of kinetic energy the parcel gains from the environment, whereas the negative area is the amount of kinetic energy that must be supplied to a parcel of air in a stable layer along an adiabat to maintain parcel ascent. Both positive area (PBE) and negative area values are output from the PBE program, and either represents the amount of potential buoyant energy in the sounding or the energy needed to make the sounding more potentially buoyant. For purposes of this research, PBE (positive area) is the

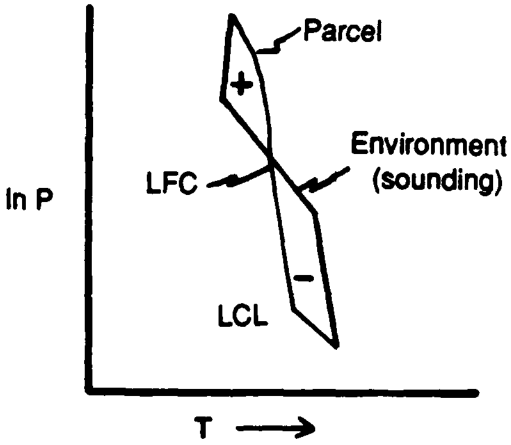


Figure 2.8. Positive area of a sounding defining Potential Buoyant Energy (PBE).

parameter of interest supplemented by low-level vertical wind shear to classify tornadic storms.

2.4 Description of Regression Techniques

Data from each outbreak case were plotted. Specifically, UMAX and MDA were plotted against each other with isolines of F_w superposed upon the plot. The plots were used to indicate what values of UMAX and MDA were needed to characterize a storm as tornadic or non-tornadic, and if so, how intense it would become. In addition, values of PBE and shear were plotted against each other in hopes of differentiating tornadic from non-tornadic storms.

2.4.1 Analysis of Plotted Rising-Ridge Surfaces

When a cell's UMAX and MDA were plotted against each other with respect to that cell's intensity, namely F_w, a standard surface arised. Figure 2.9 illustrates this relationship, called a "rising-ridge" surface (Box and Draper, 1987). This "rising-ridge" surface has a separation point, called a breakpoint value, that distinguishes cells from being tornadic from non-tornadic. As can be seen in figure 2.9, the tornadic intensity of the storms increase as one ascends the ridge. In this case, a separation point between tornadic and non-tornadic falls where UMAX equals 16 and MDA equals 20. In other words, any value of UMAX greater than or equal to 16 <u>and</u> MDA greater than or equal to 20 will determine a cell's behavior as being tornadic. In fact, depending on where any point falls on the surface one can find that cell's tornadic intensity as it pertains to that cell's breakpoint value. The lines of tornadic intensity (F_w) were drawn subjectively to

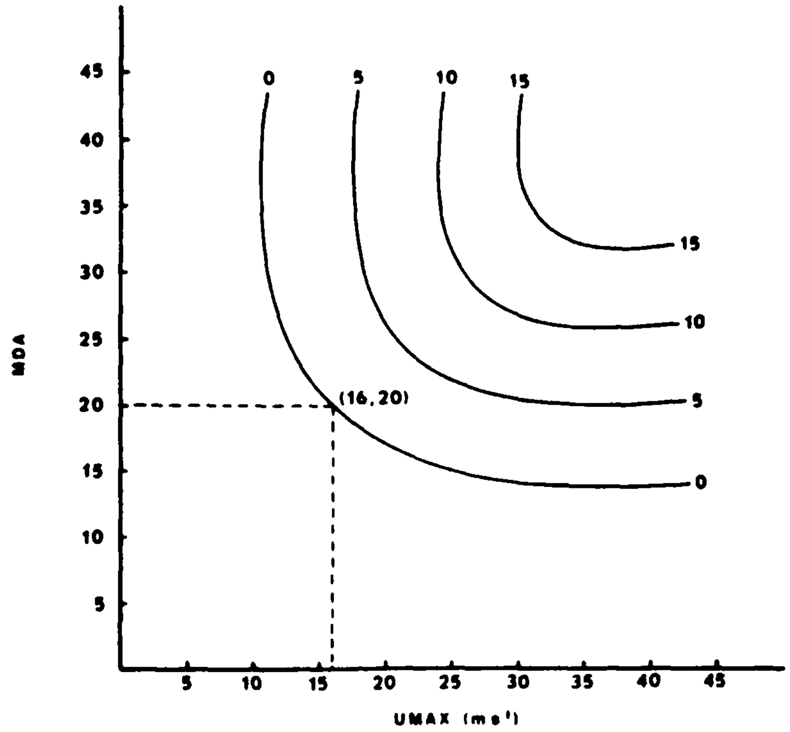


Figure 2.9. A "rising-ridge" surface defined by UMAX, MDA and F_W. The separation point, or breakpoint value, of UMAX=16 and MDA=20 defines the cutoff values needed for UMAX and MDA to fall on the "tornadic" surface.

the data points, trying to give a best fit of the lines to the data. These "eyeballed" lines describe the rising-ridge surface of the outbreak.

As a result, a different breakpoint value arose for each outbreak case. Consequently each outbreak's surface was oriented in space differently from the others, hence the nine cases had different breakpoint values. These different breakpoints led to correlation problems when all nine cases were combined into one grouped data set. In fact, when applied to an operational forecast scheme, one would have to know, a priori, the breakpoint value of a given storm in order to arrive at some prediction as to a cell's tornadic potential. Since we do not know the breakpoint of a given outbreak until after it is over, this scheme cannot be operationally used to predict the tornadic intensities of thunderstorms.

2.4.2 Quadvariate Regression Model

A four-variable statistical model was developed in-order to achieve one of the research goals set forth in this study: to arrive at some prediction of the expected tornadic intensity of a given thunderstorm cell by pre-determined measurement. This model involved a quadvariate regression scheme where, in addition to UMAX and MDA as predictors, PBE and low-level wind shear were added to predict the variance in tornadic intensities among the nine data cases. This model is described by the following equations:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \varepsilon \qquad (2.5)$$

and,

$$\begin{aligned} \hat{y} = \beta_0 &+ (\beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4) + \\ &(\beta_{11} x_1^2 + \beta_{22} x_2^2 + \beta_{33} x_3^2 + \beta_{44} x_4^2) + \\ &(\beta_{12} x_1 x_2 + \beta_{13} x_1 x_3 + \beta_{14} x_1 x_4 + \\ &\beta_{23} x_2 x_3 + \beta_{24} x_2 x_4 + \beta_{34} x_3 x_4) + \varepsilon \qquad (2.6) \end{aligned}$$

where $\hat{y}$ is the response variable F_w or F_w^2, X_1, X_2, X_3 and X_4 are the four predictor variables. Equation 2.5 is the linear regression model with 5 terms in the equation, whereas equation 2.6 is the quadratic regression model with 15 terms in the equation. In this regard, quadratic regression is not applicable in the individual outbreak cases since in four of the outbreaks there are more variables (15 terms in the equation) than observations, resulting in an "overfit" of the model to the data. For the bivariate model, one would need to have 40 observations in the data set to overcome the overfit problem. Similarly, the quadvariate model would require at least 100 observations in the data set. In the remaining data cases, there were still not enough observations in the data set to justify using the quadratic model.

Quadratic regression was carried out only on the combined and stratified data sets. However, the same caveat exists: there are too few points in the data set to justify quadratic regression. As in the bivariate regression scheme, quadvariate analysis (linear and quadratic) was also carried out on the non-dimensionalized and stratified data sets in an attempt to improve R^2, the coefficient of determination.

2.4.3 Stratification Technique

Since the grouping of the data into one data set did not significantly improve the coefficient of determination even after non-dimensionalization, another method was developed to eliminate the individuality of outbreak breakpoint values. Since each outbreak case had a different surface, and consequently, a different breakpoint, a method of grouping the outbreak cases by stratification was developed. This was done by computing the distance of each outbreak's breakpoint value from the origin. This distance was determined from the following formula:

$$d = (UMAX_{wp}^2 + MDA_{wp}^2)^{1/2} \qquad \{2.7\}$$

Once the nine d-values were obtained, the cases were then grouped by how close each case's d-value was to each other's. This was then applied to multivariate regression models to assess how well the grouped data sets fit the models.

3. RESULTS OF STRATIFICATION TECHNIQUE

The analyses results for UMAX, MDA, shear and PBE are shown in appendix 6.5. It contains the following information about each cell per outbreak: observation time, upper air station(s) used in the shear and PBE analysis, maximum radial velocity (UMAX), measured deviation angle (MDA), low-level vertical wind shear and potential buoyant energy (PBE). In all but the North Carolina case, 1200Z soundings were used.

3.1 Analysis of UMAX and MDA to F_w

When the relationship between UMAX and MDA were plotted as a function of tornadic intensity, all nine outbreaks exhibited the rising-ridge pattern, but with different orientations with respect to one another. Figures 3.1 through 3.9 depict the plots of UMAX and MDA with respect to F_w. As a result of the disparity in spatially oriented plots, each outbreak resulted in a different breakpoint. Table 3.1 lists each outbreak's breakpoint and d-value.

Table 3.1. Breakpoint and D-Values of Each Tornado Outbreak

Outbreak Case		Breakpoint Value		Outbreak
Study		UMAX	MDA	d-Value
OK	(84117)	12	15	19.21
WI-IL	(84118)	14	28	31.30
IA	(84159)	15	8	17.00
NE	(85130)	10	7	12.21
OH-PA-NY	(85150)	17	25	30.23
SD	(86209)	10	19	21.47
KS	(86261)	11	7	13.04
TX-LA	(87319)	15	14	20.52
NC-VA	(88333)	6	12	13.40

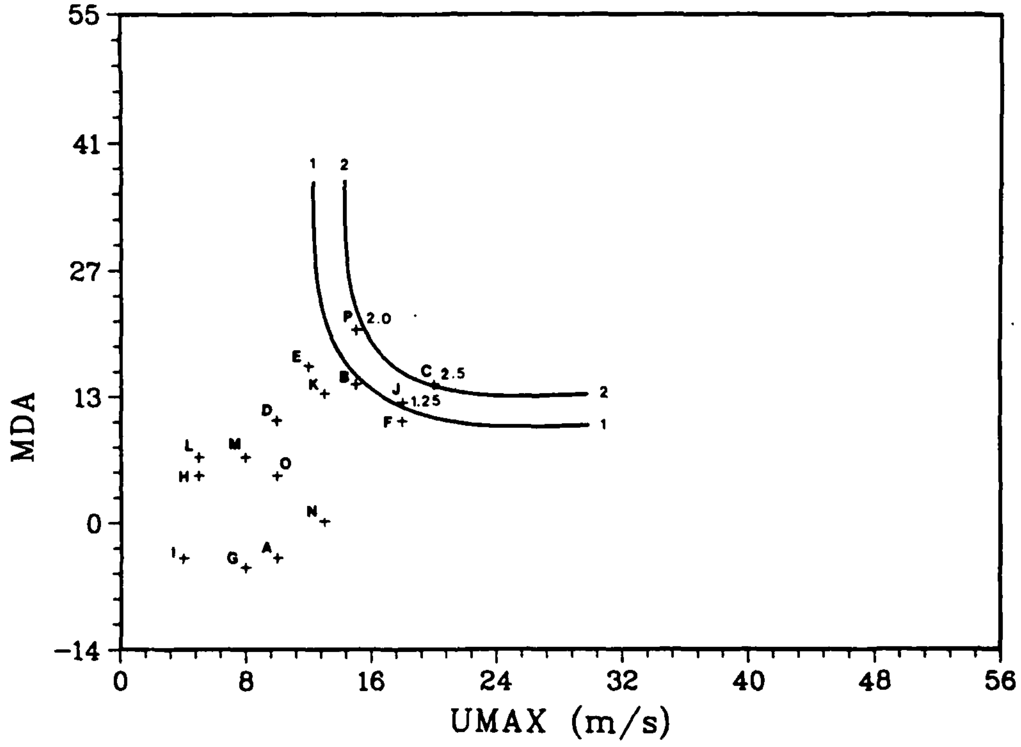


Figure 3.1. Rising-ridge plot analysis of UMAX, MDA versus F_W for the Oklahoma outbreak of 26 April, 1984. Letters indicate the cell of the outbreak. Values are the tornadic intensities of the associated cells (F_W). Cells without values were non-tornadic.

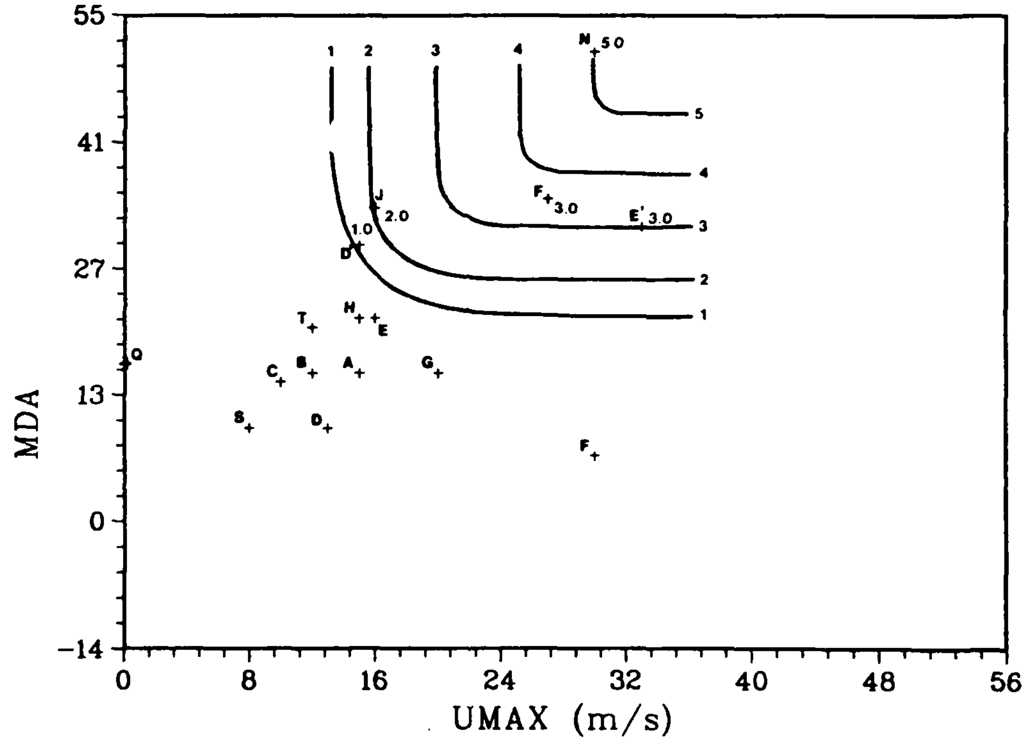


Figure 3.2. Rising-ridge plot analysis of UMAX, MDA versus F_W for the Wisconsin-Illinois outbreak of 27 April, 1984. Letters indicate the cell of the outbreak. Values are the tornadic intensities of the associated cells (F_W). Cells without values were non-tornadic.

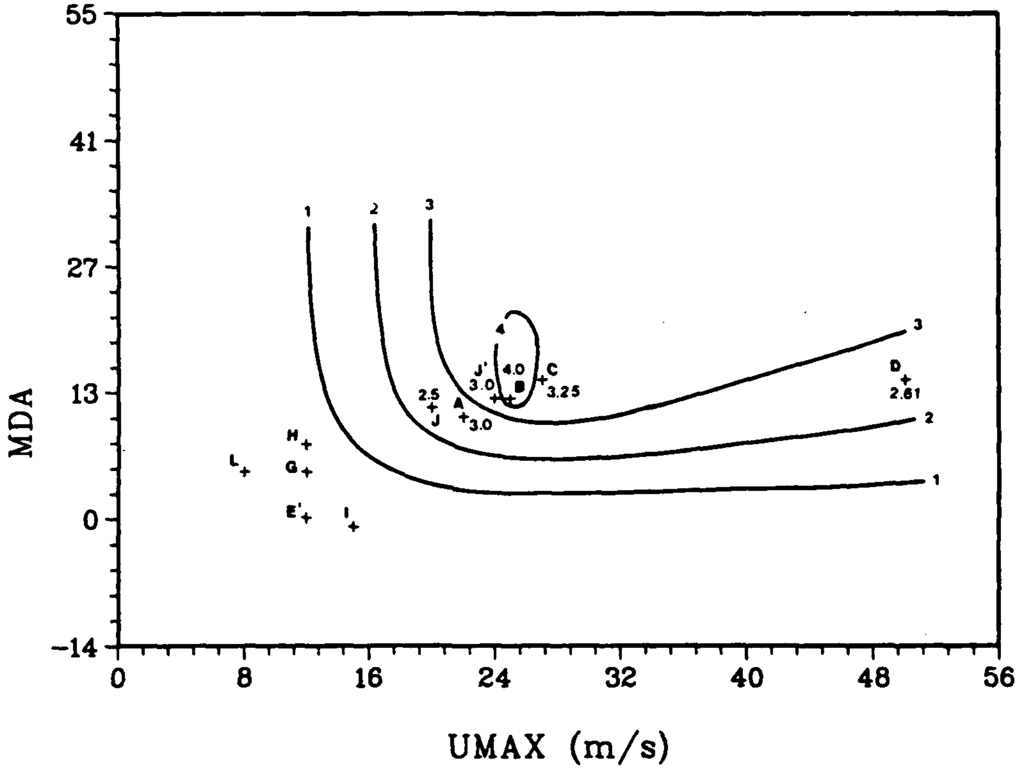


Figure 3.3. Rising-ridge plot analysis of UMAX, MDA versus F_W for the Iowa outbreak of 7 June, 1984. Letters indicate the cell of the outbreak. Values are the tornadic intensities of the associated cells (F_W). Cells without values were non-tornadic.

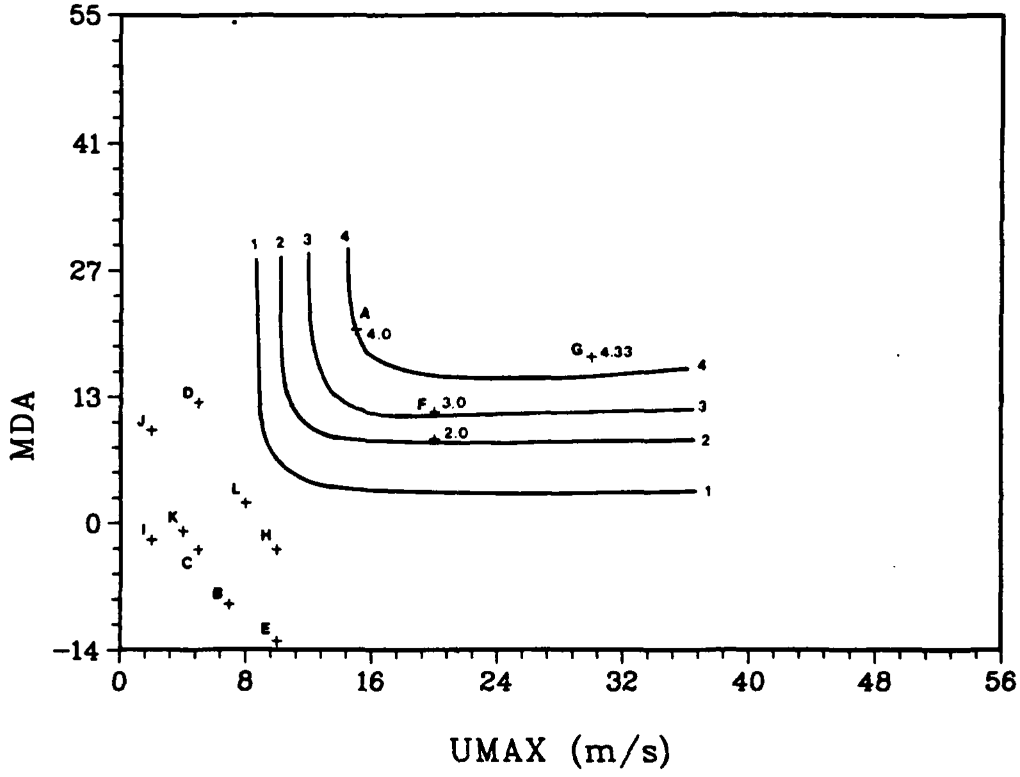


Figure 3.4. Rising-ridge plot analysis of UMAX, MDA versus F_W for the Nebraska outbreak of 10 May, 1985. Letters indicate the cell of the outbreak. Values are the tornadic intensities of the associated cells (F_W). Cells without values were non-tornadic.

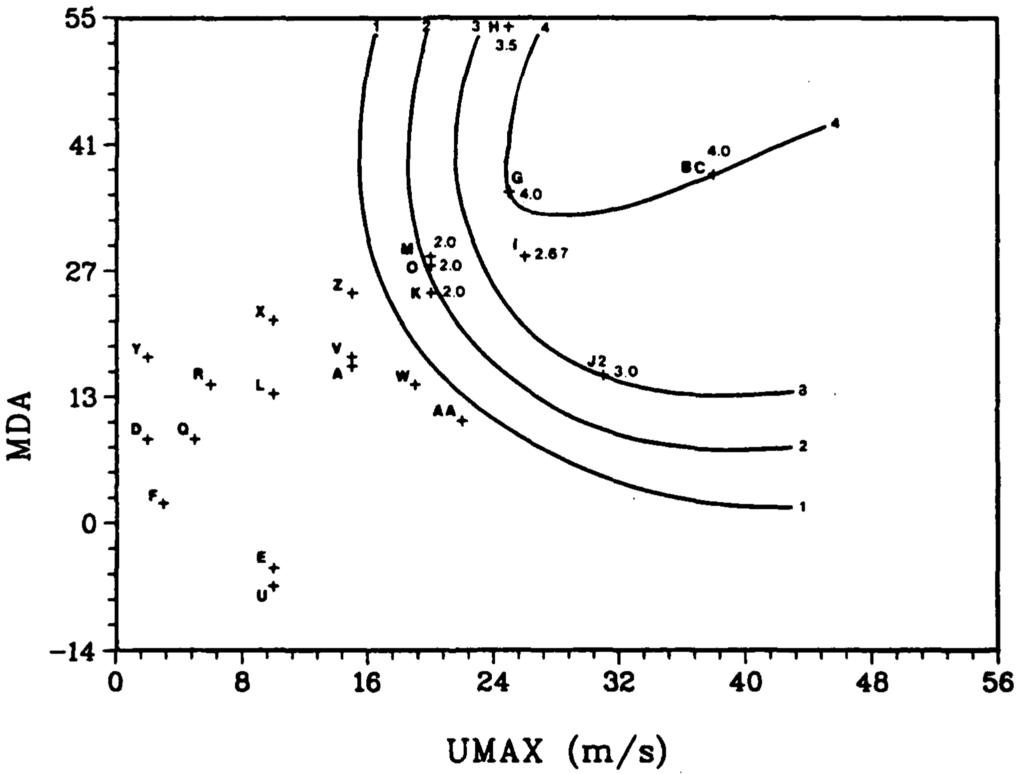


Figure 3.5. Rising-ridge plot analysis of UMAX, MDA versus F_W for the Ohio-Pennsylvania-New York outbreak of 31 May, 1985. Letters indicate the cell of the outbreak. Values are the tornadic intensities of the associated cells (F_W). Cells without values were non-tornadic.

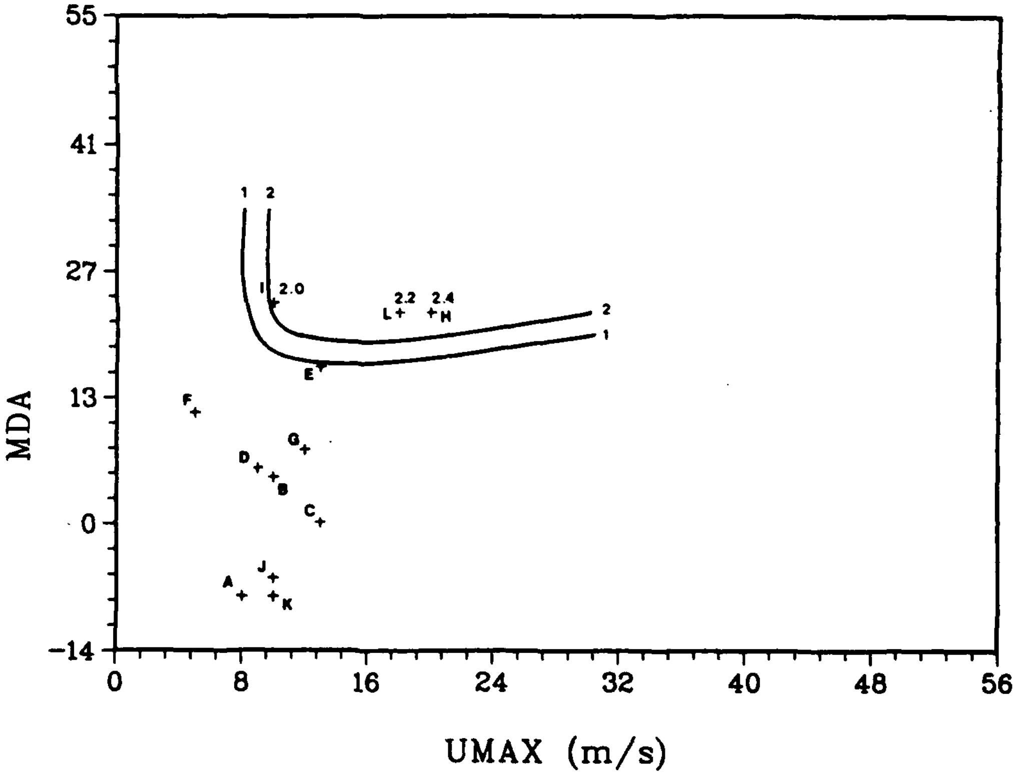


Figure 3.6. Rising-ridge plot analysis of UMAX, MDA versus F_W for the South Dakota outbreak of 28 July, 1986. Letters indicate the cell of the outbreak. Values are the tornadic intensities of the associated cells (F_W). Cells without values were non-tornadic.

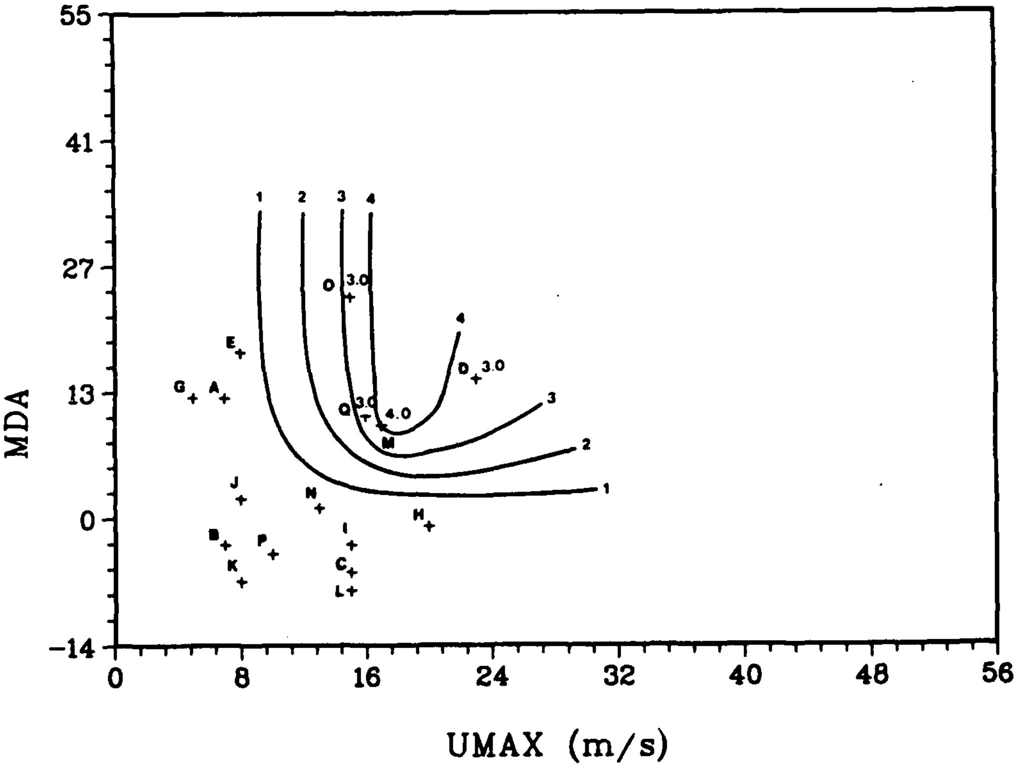


Figure 3.7. Rising-ridge plot analysis of UMAX, MDA versus F_W for the Kansas outbreak of 18 September, 1986. Letters indicate the cell of the outbreak. Values are the tornadic intensities of the associated cells (F_W). Cells without values were non-tornadic.

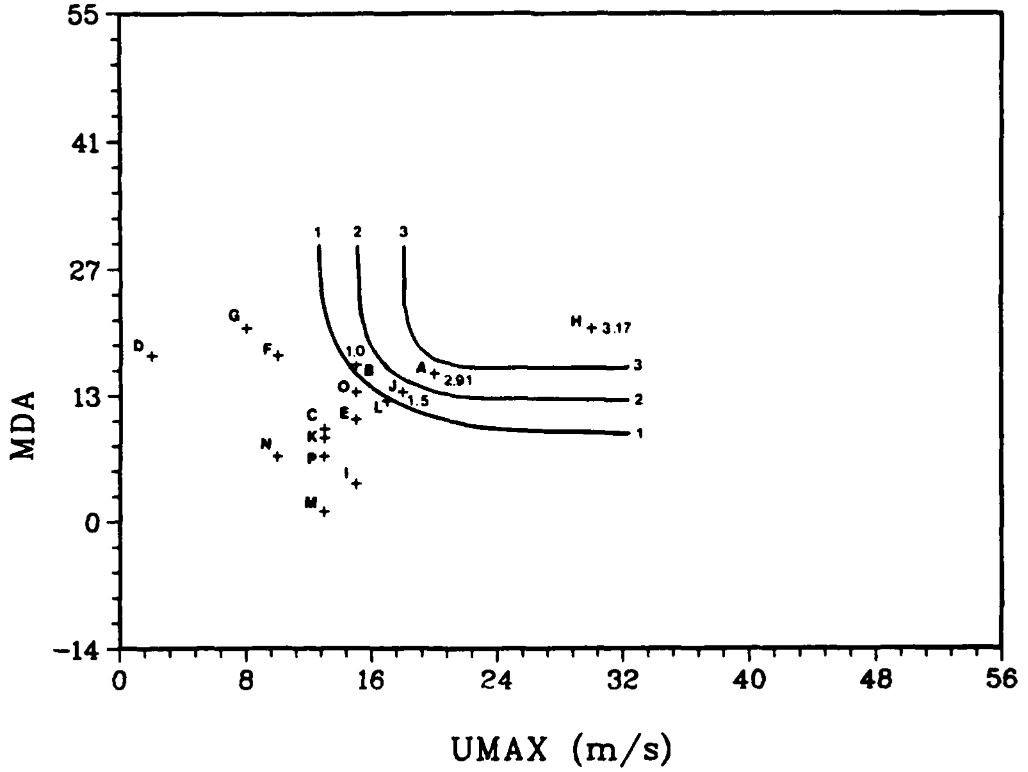


Figure 3.8. Rising-ridge plot analysis of UMAX, MDA versus F_W for the Texas-Louisiana outbreak of 17 November, 1987. Letters indicate the cell of the outbreak. Values are the tornadic intensities of the associated cells (F_W). Cells without values were non-tornadic.

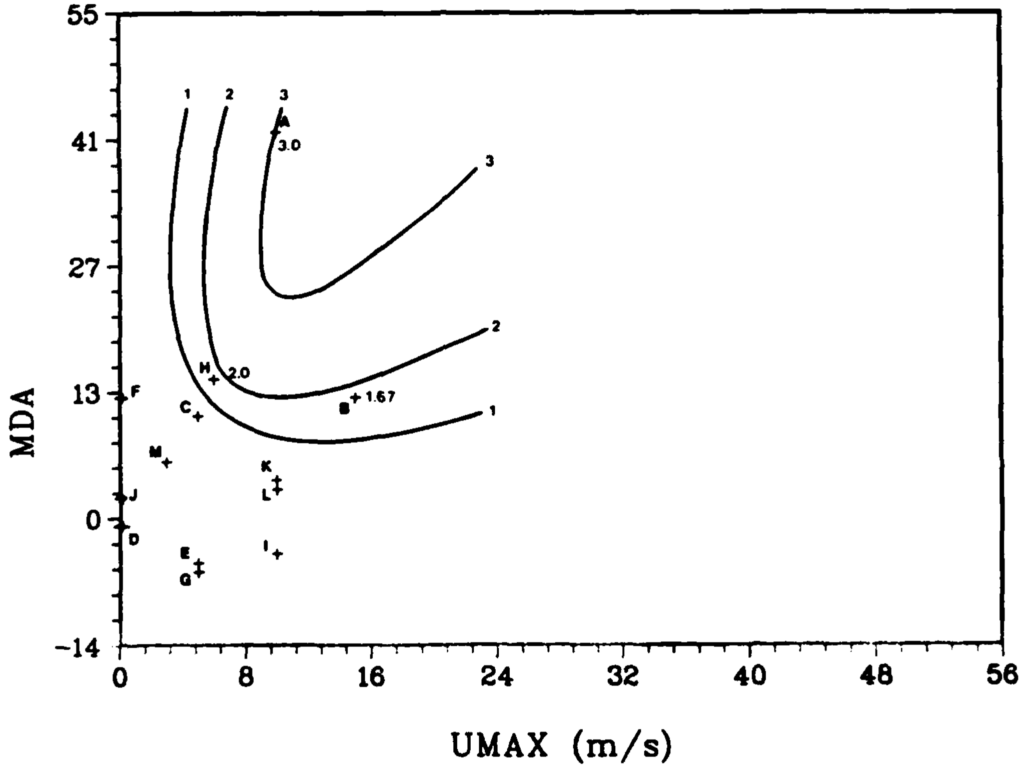


Figure 3.9. Rising-ridge plot analysis of UMAX, MDA versus F_w for the North Carolina-Virginia outbreak of 28 November, 1988. Letters indicate the cell of the outbreak. Values are the tornadic intensities of the associated cells (F_w). Cells without values were non-tornadic.

The isolines of F_w were subjectively drawn to fit the values of F_w to the surface of the plot. This "eyeballing" of the isolines to the data does not lend itself to scientific accuracy. It is only a descriptive analysis of how UMAX and MDA relate to tornadic intensity.

The breakpoint value of each outbreak case is based on the point between the "1" isoline and zero-values of F_w. In some cases (notably the OK and WI-IL outbreaks) an average breakpoint was determined when there was a large space between the non-occurances and the F0 line. Note that the "1" isoline represents the beginning of tornado activity, or F0 scale, with each successive value of isoline corresponding to the F_w scale.

In order to determine which method of classifying tornadic intensities was best, i.e. F_w or F_w^2, bivariate regression was performed on the nine individual cases involving the predictor variables UMAX and MDA. Linear regression analysis was carried out using F_w and F_w^2 as the response variables. The results of these analyses are shown in Table 3.2. These results indicate that the

Table 3.2. Percent of variation (coefficient of determination or R^2) in tornadic intensity (F_w and F_w^2) that can be explained by the predictor variables UMAX and MDA for linear bivariate regression.

			R^2	
Case		FW	FW²	Difference
OK	(84117)	0.4065	0.3680	0.0385
WI-IL	(84118)	0.8948	0.8031	0.0917
IA	(84159)	0.7214	0.6412	0.0802
NE	(85130)	0.8861	0.8270	0.0591
OH-PA-NY	(85150)	0.7658	0.7267	0.0391
SD	(86209)	0.7058	0.7268	-0.0210
KS	(86261)	0.6326	0.5531	0.0795
TX-LA	(87319)	0.7193	0.7490	-0.0297
NC-VA	(88333)	0.7724	0.8254	-0.0530

linear bivariate regression has highly variable results from case to case and that F_w is a better indicator of tornadic intensity than F_w^2, in most of the outbreak cases. On the average, F_w accounted for 72% of of the variance in tornadic intensity by UMAX and MDA, whereas F_w^2 accounted for just over 69% of the variance.

One reason F_w was a better response variable than F_w^2 lies in the data. By looking at any of the plotted data of UMAX and MDA to F_w and F_w^2, one observes there are more zero values than non-zero values. The statistical distribution of this and any other outbreak population tends to put the non-zero values in the tails of the distribution curve whereas the zeroes appear near the mean in a normal population. In terms of using F_w versus F_w^2, F_w values are not as large as the F_w^2 results, and consequently do not embody as large a data distribution between zero and non-zero values. In other words, F_w values are not as prone to appear in the extreme regions of the tails as F_w^2 values tend to be, thereby eliminating some of the error in the statistical analysis.

Figures 3.10 through 3.13 show this distinction. These figures are three-dimensional plots of the predictor variables UMAX and MDA (x- and y-axis) to the response variable F_w or F_w^2 (z-axis). The plane that is depicted on each plot is the linear surface that best fits the data by the regression coefficients (intercept and parameter coefficients; see equation 1.1). Also plotted are the residuals of observed tornadic cells to the predicted F_w (or F_w^2) of that cell. As can be seen from the plots, the residuals are smaller in the F_w analysis than the F_w^2 analysis.

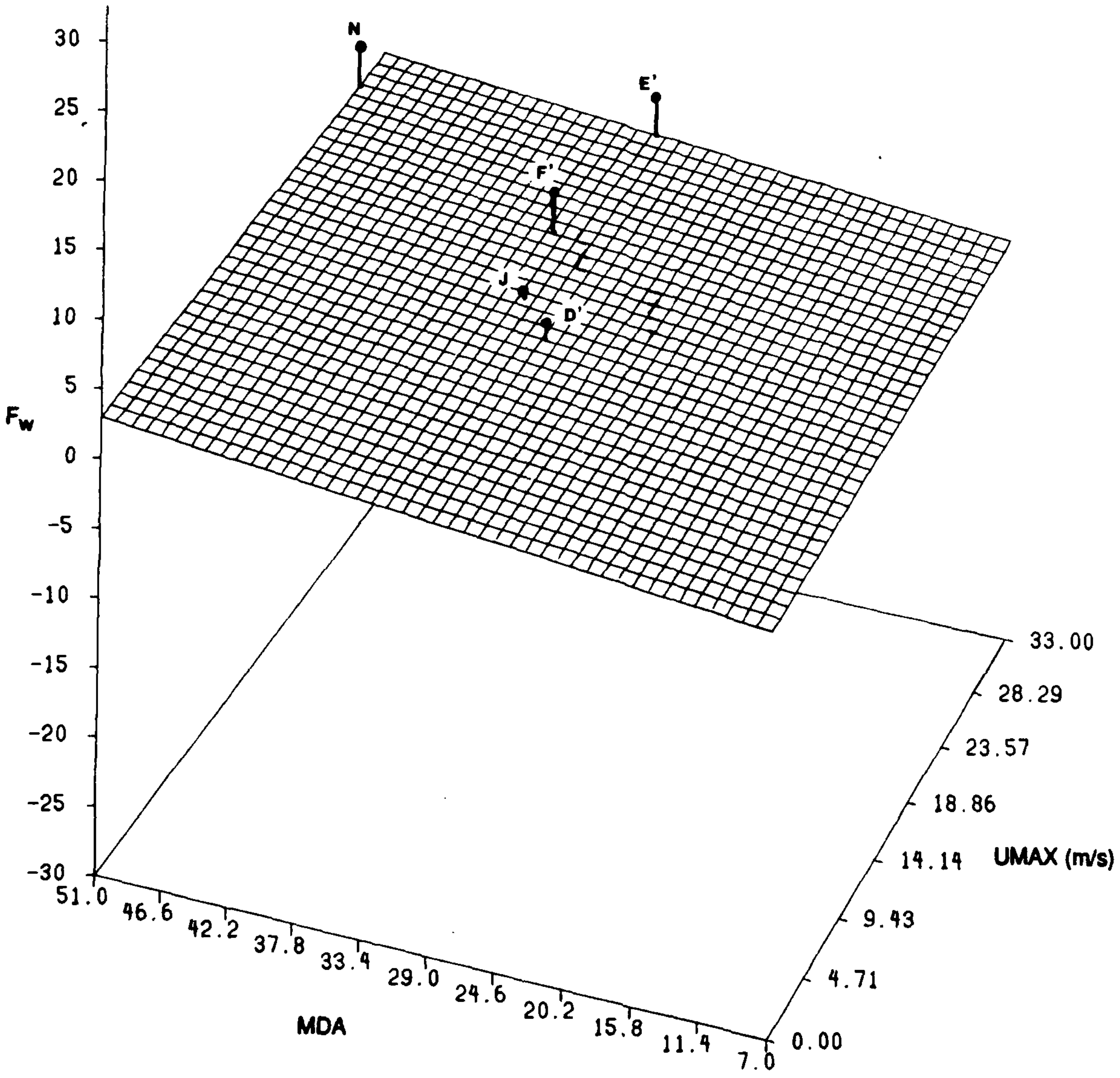


Figure 3.10. Three-dimensional plot of the regression analysis for the WI-IL outbreak. The predictor variables UMAX (x-axis) and MDA (y-axis) were plotted against the response variable F_W (z-axis). The letters above the linear regression plane were the tornadic cells actually observed for UMAX, MDA and F_W, with the shown departures of these cells from the plane.

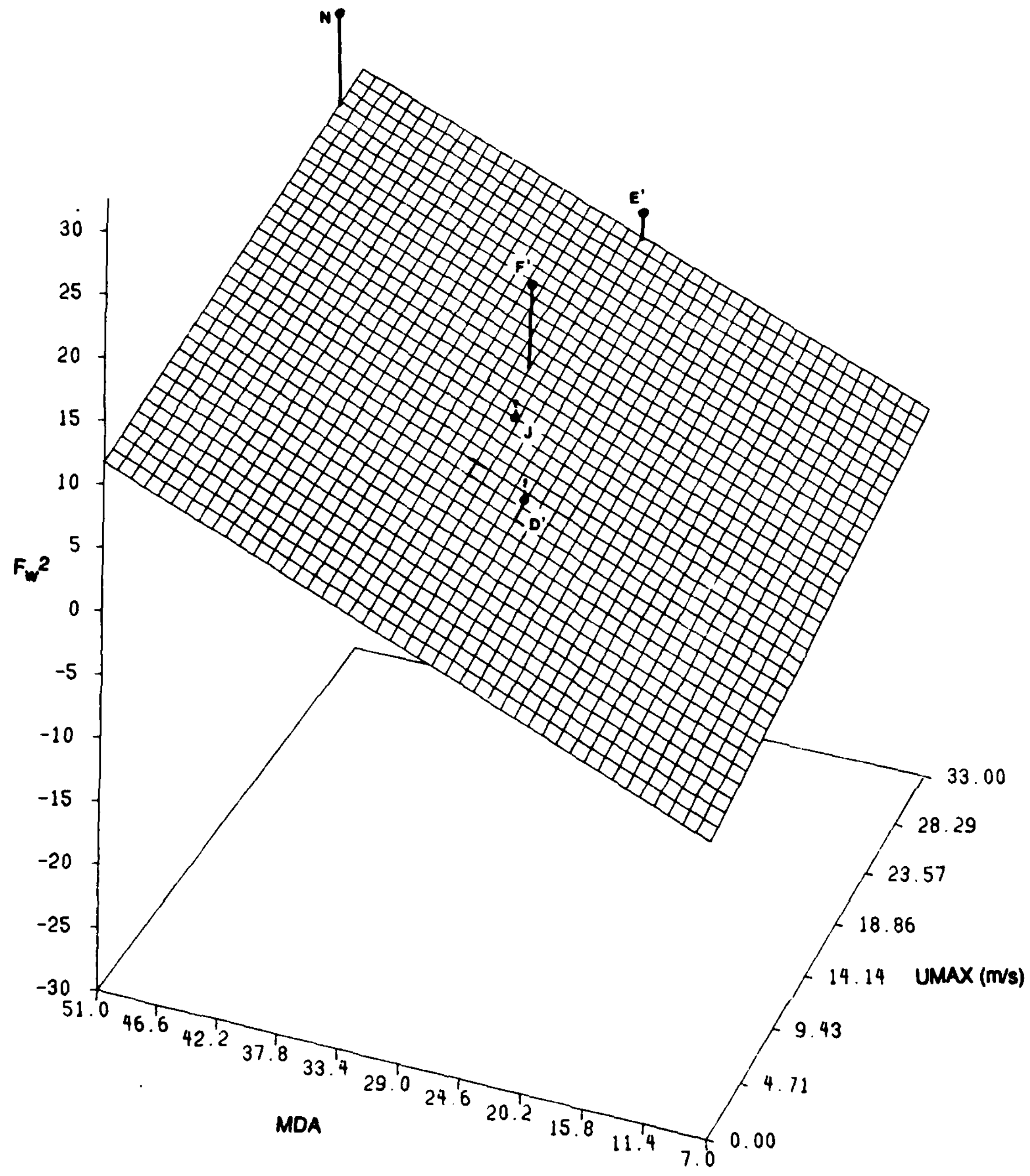


Figure 3.11. Three-dimensional plot of the regression analysis for the WI-IL outbreak. The predictor variables UMAX (x-axis) and MDA (y-axis) were plotted against the response variable F_w^2 (z-axis). The letters above and below the linear regression plane were the tornadic cells actually observed for UMAX, MDA and F_w^2, with the shown departures of these cells from the plane.

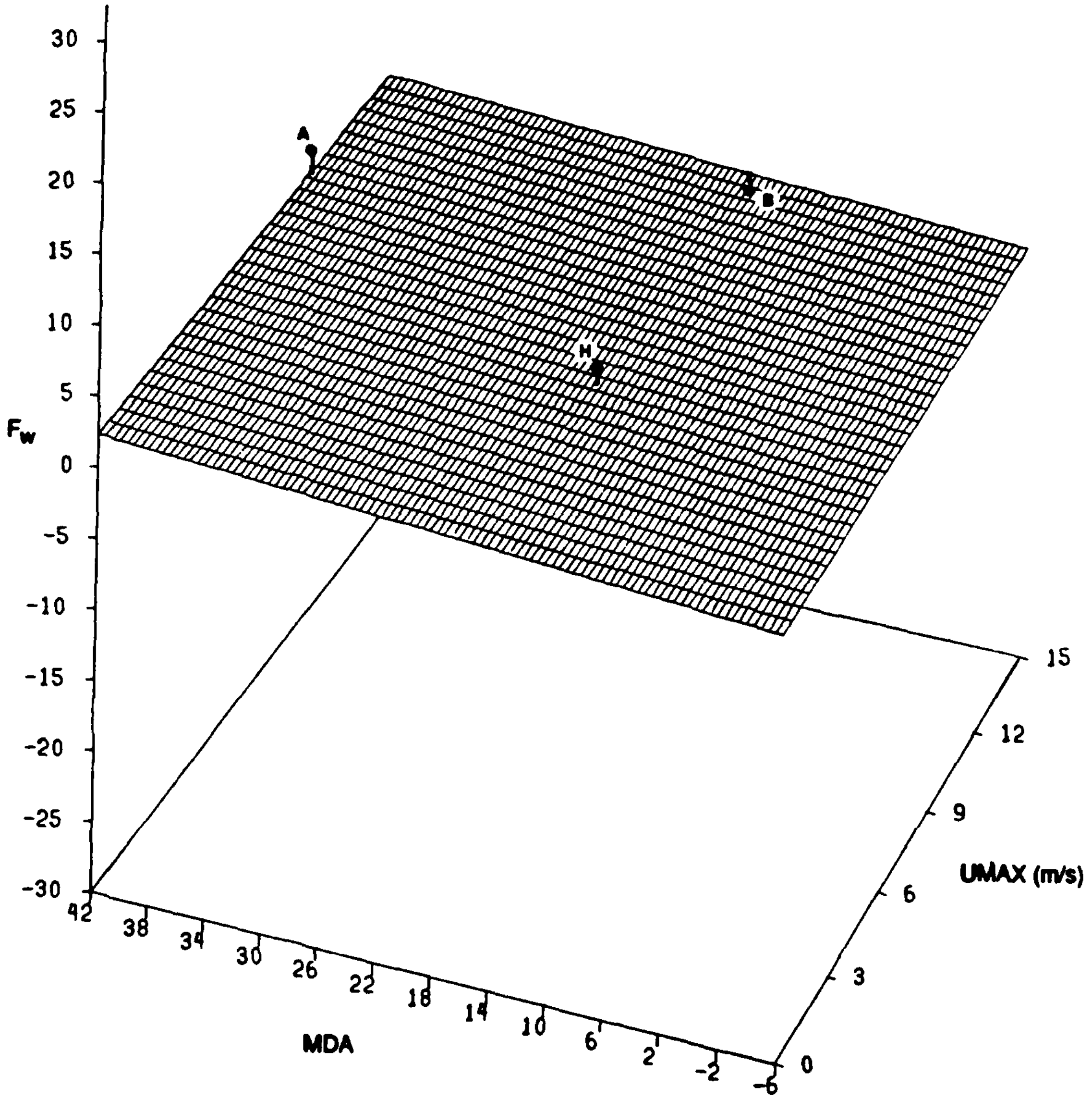


Figure 3.12. Three-dimensional plot of the regression analysis for the NC-VA outbreak. The predictor variables UMAX (x-axis) and MDA (y-axis) were plotted against the response variable F_w (z-axis). The letters above and below the linear regression plane were the tornadic cells actually observed for UMAX, MDA and F_w, with the shown departures of these cells from the plane.

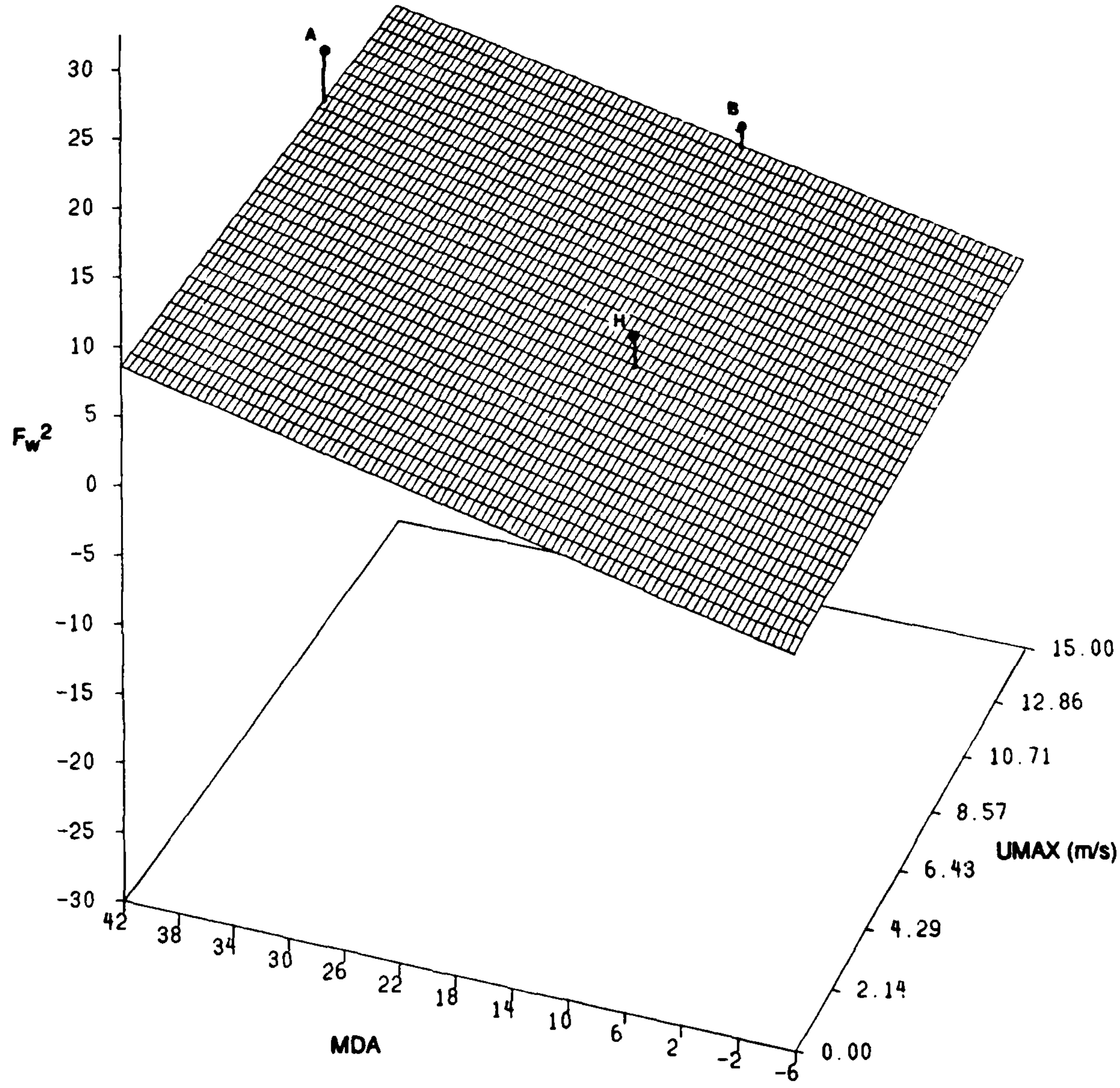


Figure 3.13. Three-dimensional plot of the regression analysis for the NC-VA outbreak. The predictor variables UMAX (x-axis) and MDA (y-axis) were plotted against the response variable F_w^2 (z-axis). The letters above the linear regression plane were the tornadic cells actually observed for UMAX, MDA and F_w^2, with the shown departures of these cells from the plane.

3.2 Analysis of Shear and PBE to FW

Shear and PBE were plotted against each other to see if they both exhibited a function of tornadic or non-tornadic distinction for each outbreak. By utilizing the same distinction line between tornadic and non-tornadic occurrences used by Rasmussen and Wilhelmson in 1983, each outbreak's observations were plotted to see how they fit the graph (figures 3.14 - 3.22). The results were promising, but some problems became evident. The Iowa outbreak did not show this distinction very well for tornadic cells B, C, J and J' as they fell in the non-tornadic region of the graph. Cell G fell in the meso-cyclone part of the graph, and cell D fell in the tornadic area. The Kansas outbreak did a little better, with two cells falling in the extreme tornadic area and two in the extreme non-tornadic region. These two "non-tornadic" cells developed in Colorado, and cells that developed there tend to be of a different variety, as PBE is not a very good indicator of instability in the high plains of the west (Rasmussen, 1988). The Wisconsin, South Dakota and Texas outbreaks showed excellent fits to the graph. Most of the others had good fits as well, but some cells fell in the meso-cyclone region and some fell in the tornadic region of the graph.

However, the Ohio-Pennsylvania-New York outbreak failed to show any connection of PBE and shear to tornadic activity. Not only were there no cells in the tornadic region, but none of the cells displayed very high PBE values. There was evidence, however, of strong low-level shear, particularly in the observed tornadic cells. Another thing to note is that although this was one of the largest outbreaks of the decade, PBE did not well characterize the situation. One of the

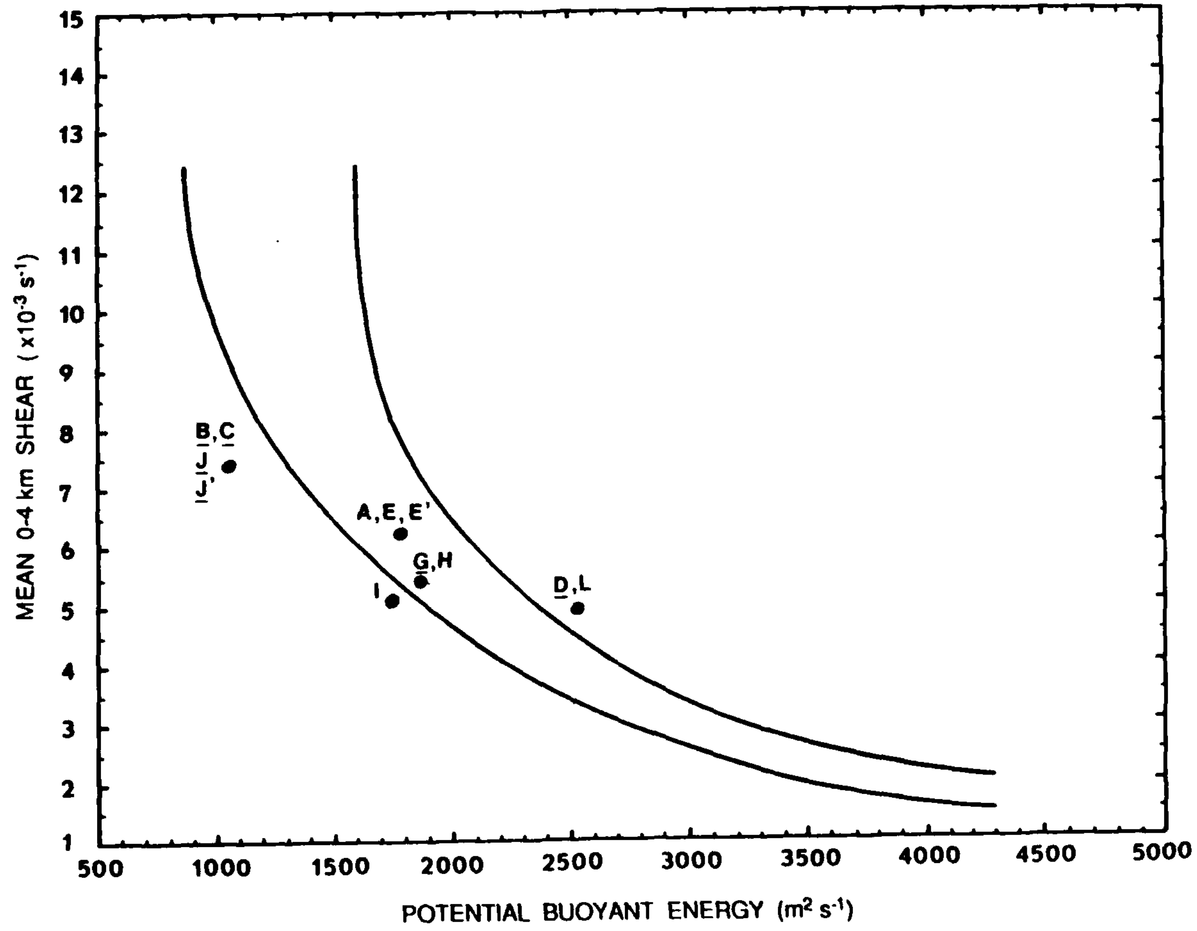


Figure 3.14. Potential Buoyant Energy (PBE) and low-level (0-4 km) vertical wind shear analysis for the IA outbreak. Letters indicate the cell of the outbreak. Underlined cells were tornadic. From Rasmussen and Wilhelmson, 1983, the solid upper line separates tornado- from mesocyclone-producing storms. The region below the lower solid line represents non-tornadic storms. (after Rasmussen & Wilhelmson, 1983).

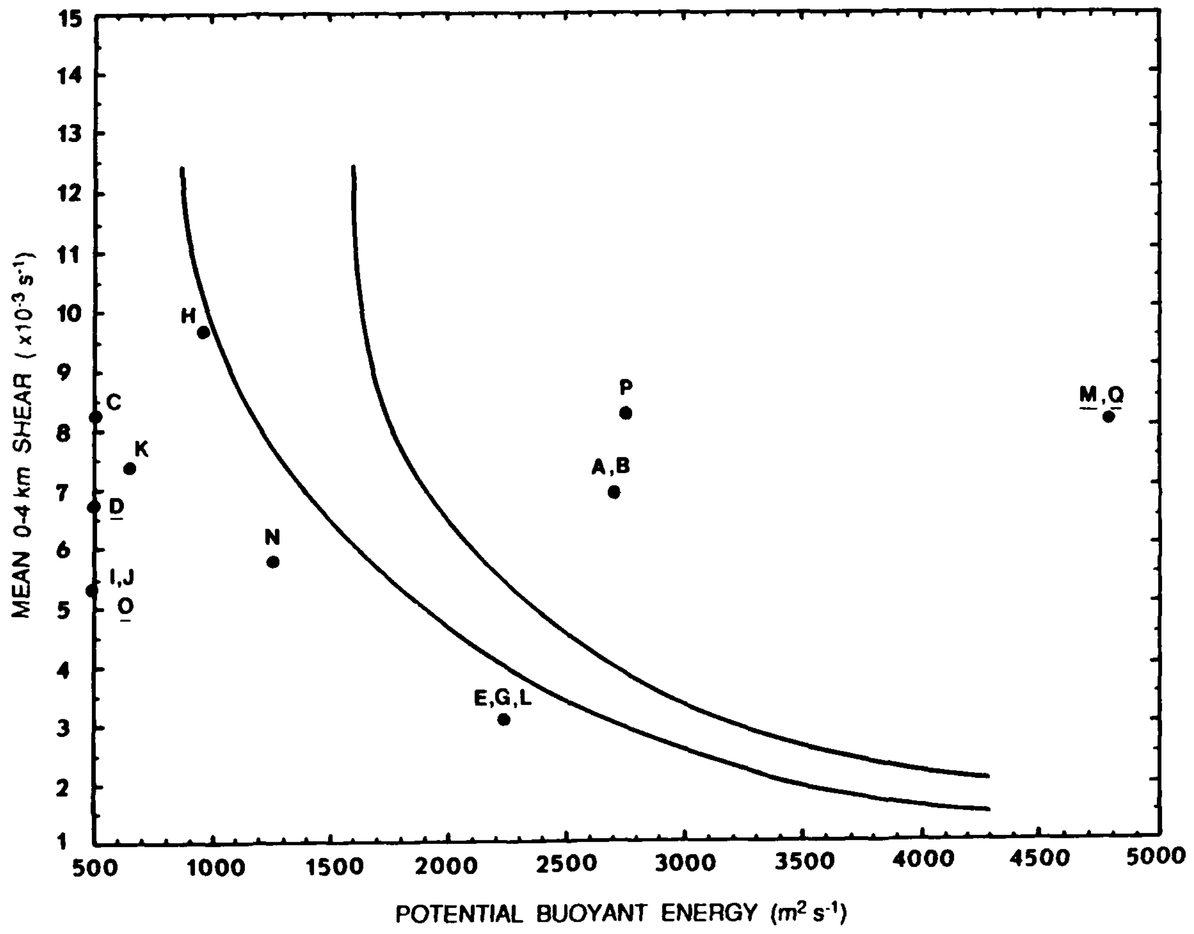


Figure 3.15. Potential Buoyant Energy (PBE) and low-level (0-4 km) vertical wind shear analysis for the KS outbreak. Letters indicate the cell of the outbreak. Underlined cells were tornadic. From Rasmussen and Wilhelmson, 1983, the solid upper line separates tornado- from mesocyclone-producing storms. The region below the lower solid line represents non-tornadic storms. (after Rasmussen & Wilhelmson, 1983).

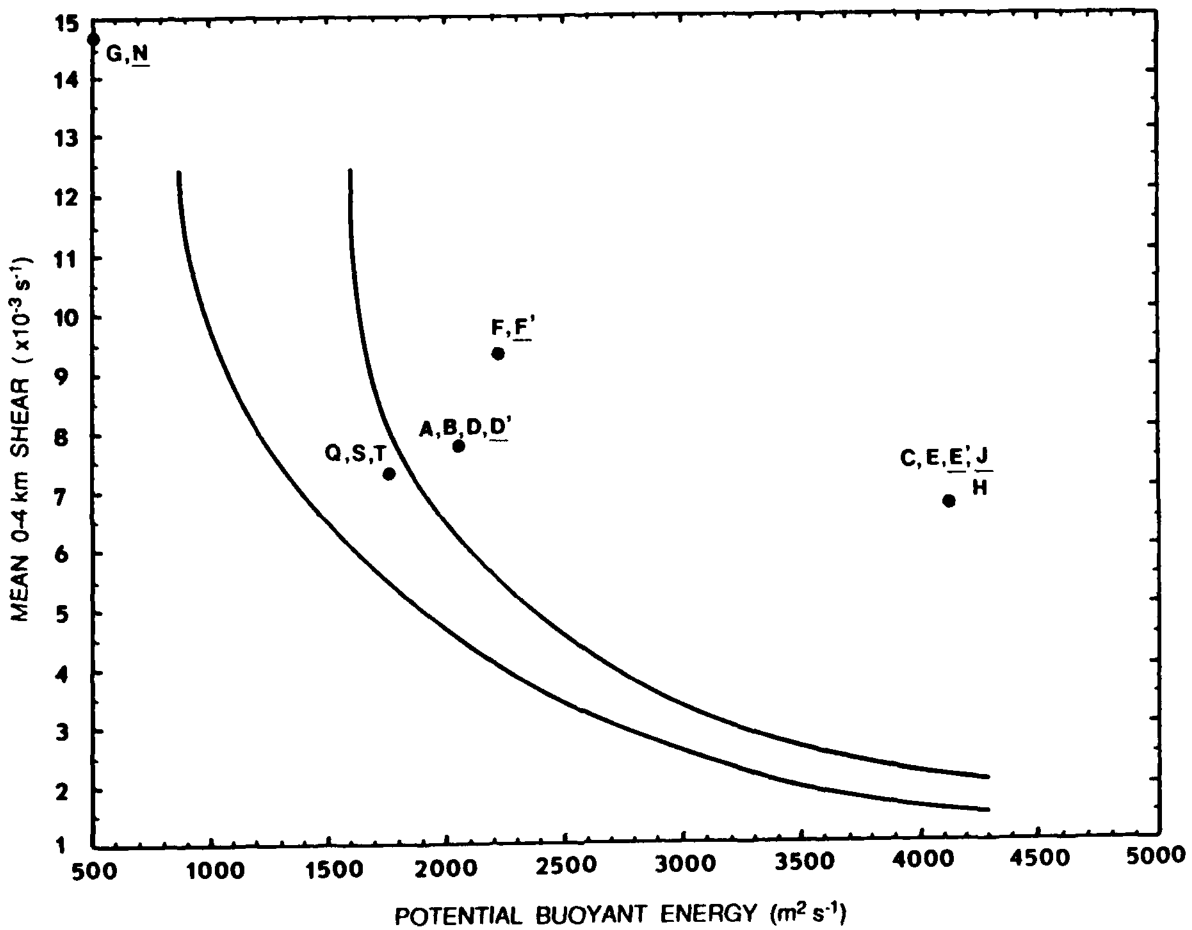


Figure 3.16. Potential Buoyant Energy (PBE) and low-level (0-4 km) vertical wind shear analysis for the WI-IL outbreak. Letters indicate the cell of the outbreak. Underlined cells were tornadic. From Rasmussen and Wilhelmson, 1983, the solid upper line separates tornado- from mesocyclone-producing storms. The region below the lower solid line represents non-tornadic storms. (after Rasmussen & Wilhelmson, 1983).

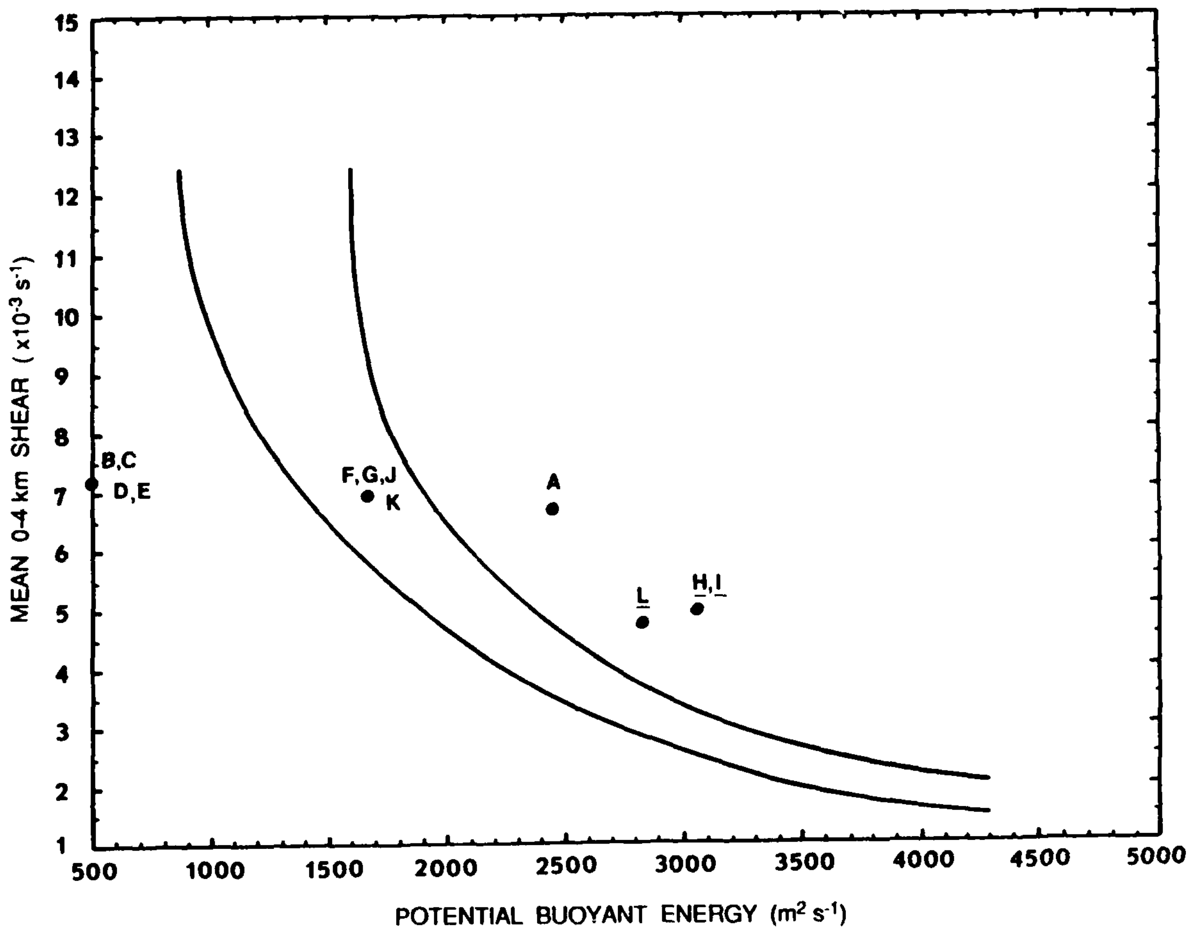


Figure 3.17. Potential Buoyant Energy (PBE) and low-level (0-4 km) vertical wind shear analysis for the SD outbreak. Letters indicate the cell of the outbreak. Underlined cells were tornadic. From Rasmussen and Wilhelmson, 1983, the solid upper line separates tornado- from mesocyclone-producing storms. The region below the lower solid line represents non-tornadic storms. (after Rasmussen & Wilhelmson, 1983).

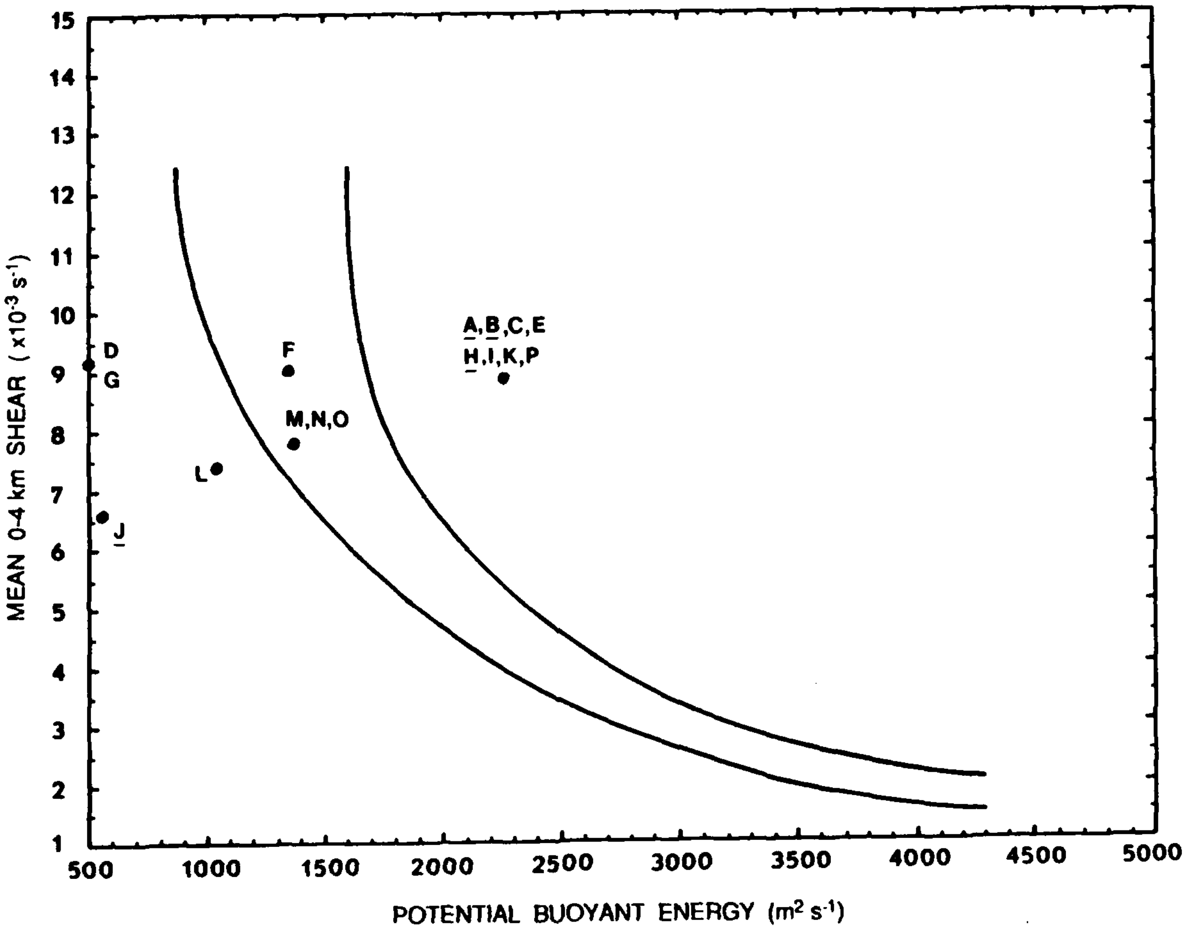


Figure 3.18. Potential Buoyant Energy (PBE) and low-level (0-4 km) vertical wind shear analysis for the TX-LA outbreak. Letters indicate the cell of the outbreak. Underlined cells were tornadic. From Rasmussen and Wilhelmson, 1983, the solid upper line separates tornado- from mesocyclone-producing storms. The region below the lower solid line represents non-tornadic storms. (after Rasmussen & Wilhelmson, 1983).

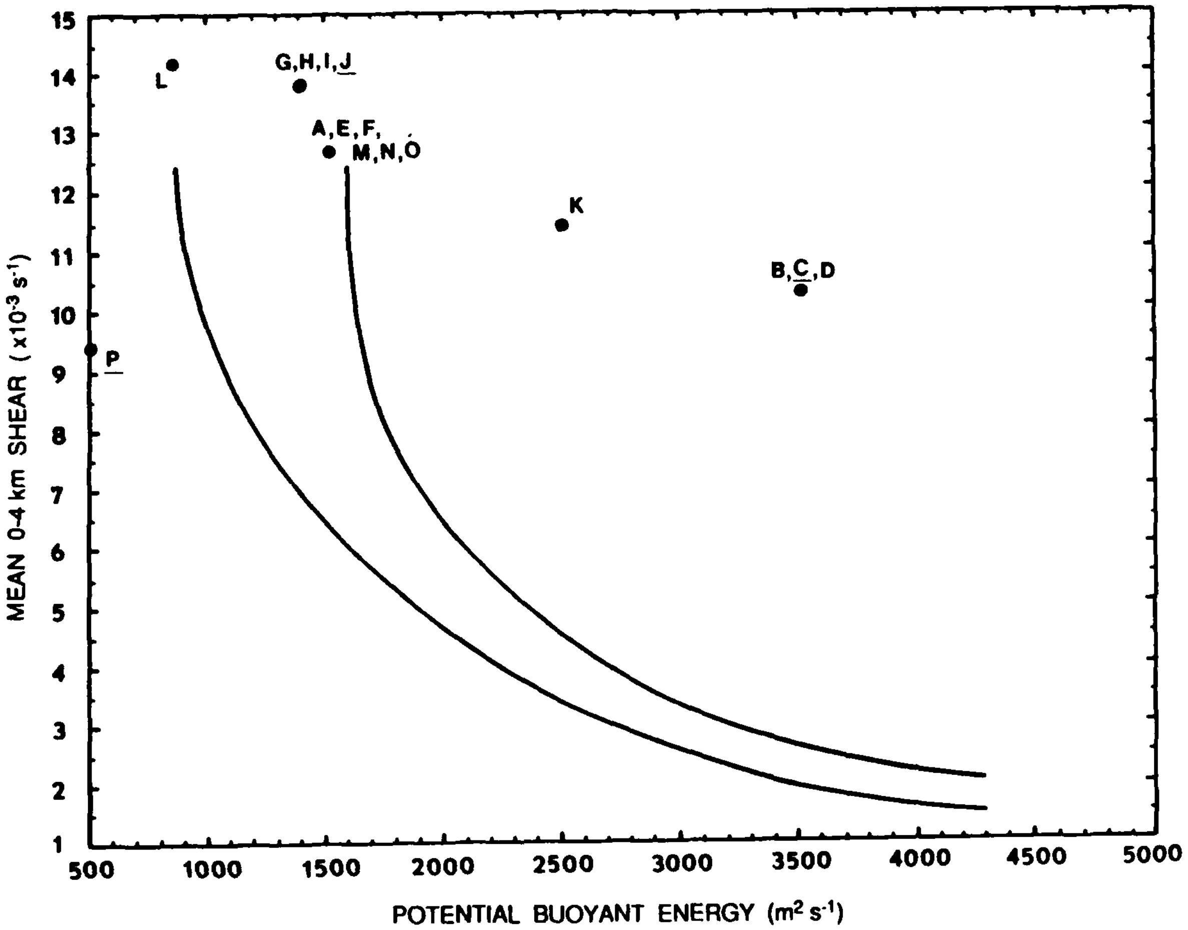


Figure 3.19. Potential Buoyant Energy (PBE) and low-level (0-4 km) vertical wind shear analysis for the OK outbreak. Letters indicate the cell of the outbreak. Underlined cells were tornadic. From Rasmussen and Wilhelmson, 1983, the solid upper line separates tornado- from mesocyclone-producing storms. The region below the lower solid line represents non-tornadic storms. (after Rasmussen & Wilhelmson, 1983).

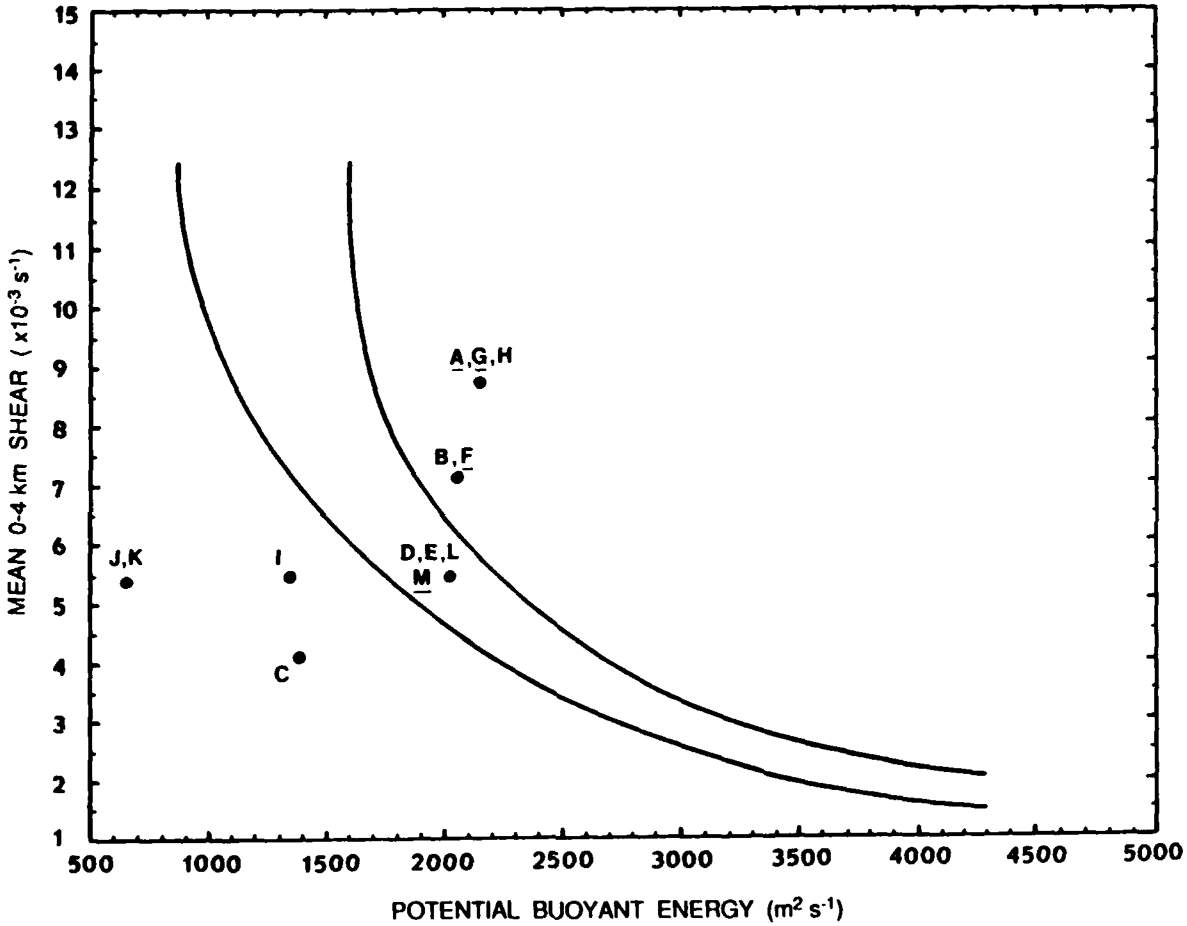


Figure 3.20. Potential Buoyant Energy (PBE) and low-level (0-4 km) vertical wind shear analysis for the NE outbreak. Letters indicate the cell of the outbreak. Underlined cells were tornadic. From Rasmussen and Wilhelmson, 1983, the solid upper line separates tornado- from mesocyclone-producing storms. The region below the lower solid line represents non-tornadic storms. (after Rasmussen & Wilhelmson, 1983).

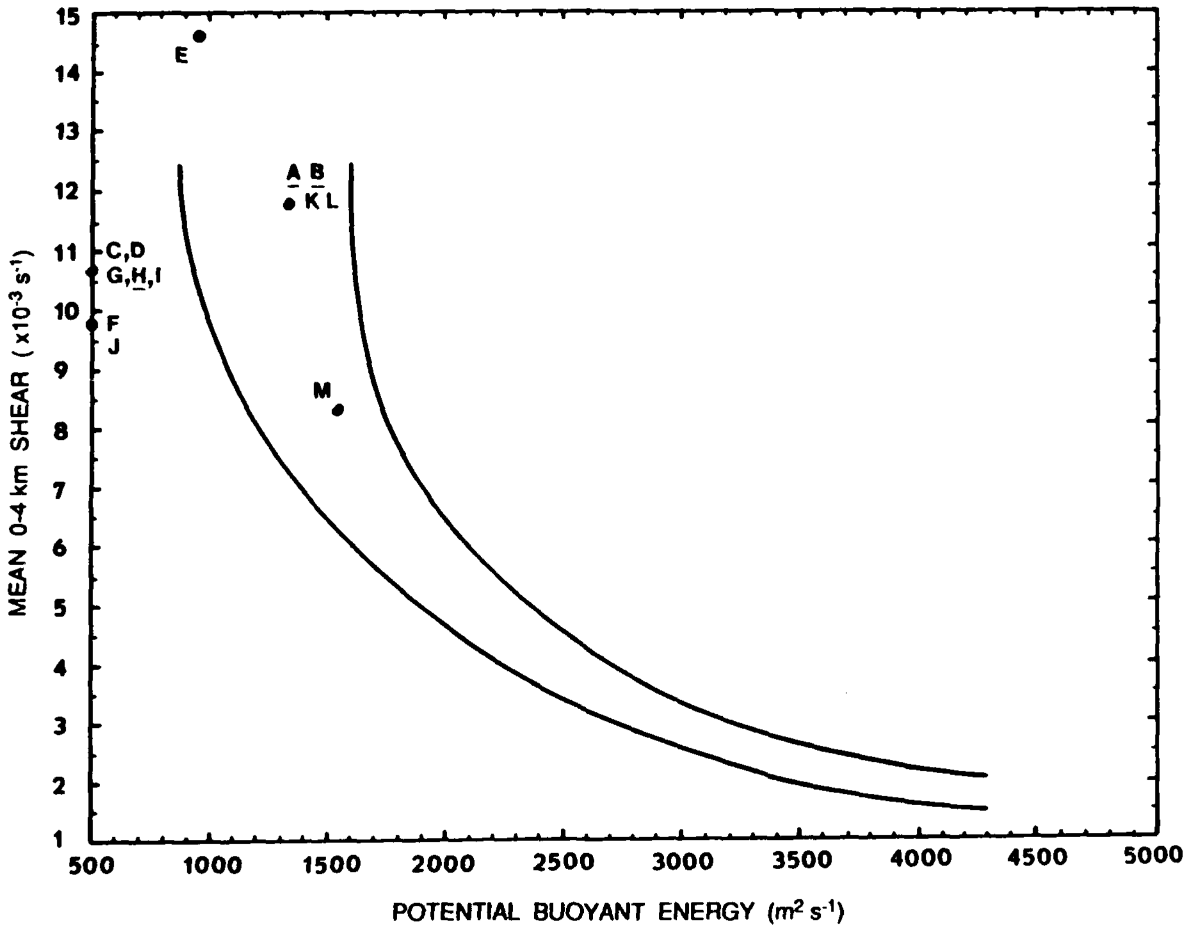


Figure 3.21. Potential Buoyant Energy (PBE) and low-level (0-4 km) vertical wind shear analysis for the NC-VA outbreak. Letters indicate the cell of the outbreak. Underlined cells were tornadic. From Rasmussen and Wilhelmson, 1983, the solid upper line separates tornado- from mesocyclone-producing storms. The region below the lcwer solid line represents non-tornadic storms. (after Rasmussen & Wilhelmson, 1983).

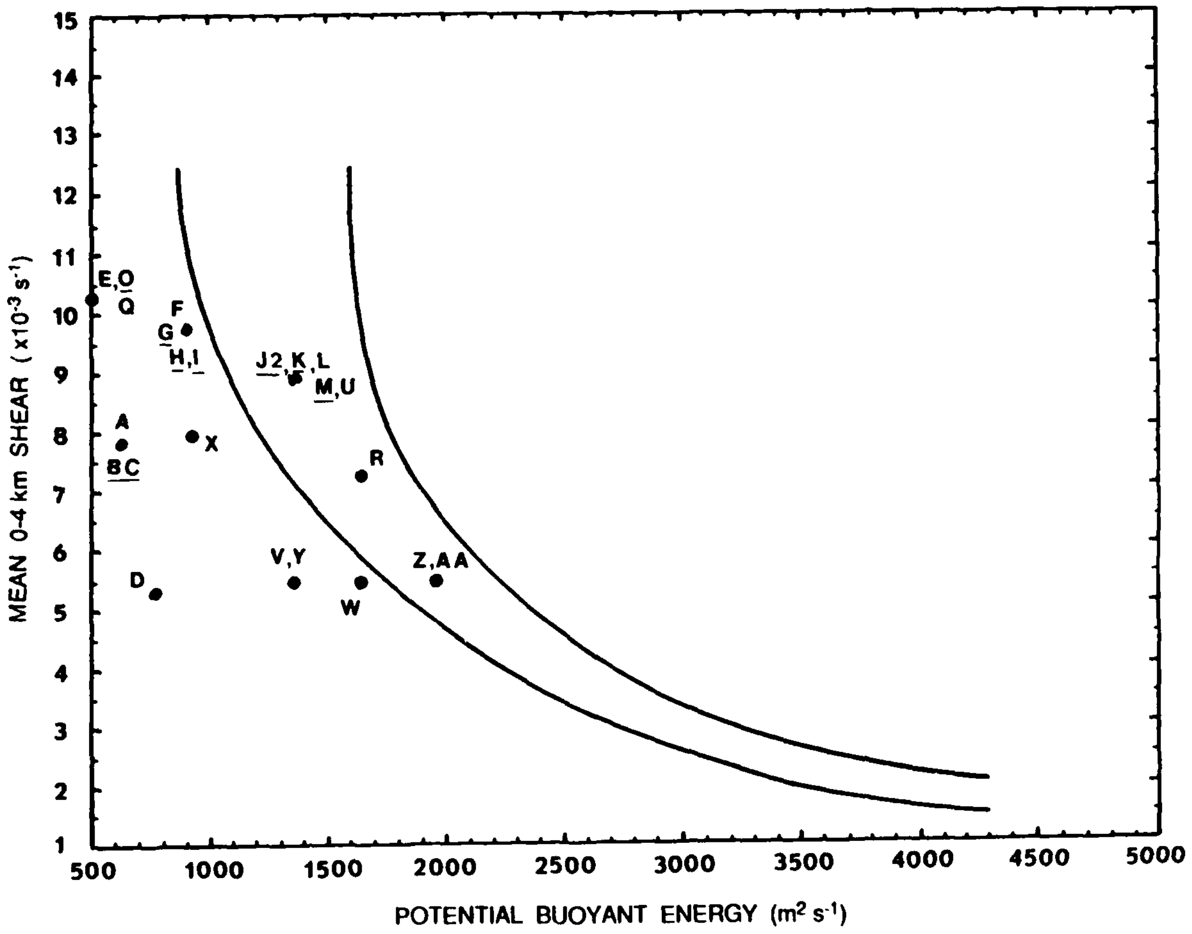


Figure 3.22. Potential Buoyant Energy (PBE) and low-level (0-4 km) vertical wind shear analysis for the OH-PA-NY outbreak. Letters indicate the cell of the outbreak. Underlined cells were tornadic. From Rasmussen and Wilhelmson, 1983, the solid upper line separates tornado- from mesocyclone-producing storms. The rgion below the lower solid line represents non-tornadic storms. (after Rasmussen & Wilhelmson, 1983).

reasons may be due to the presence of strong capping inversions in the lower layer during this outbreak, thereby "choking" off the positive area (PBE) of the sounding. However once the inversion was broken, very strong instability resulted in the very explosive development of these storms.

Bivariate linear regression was similarly carried out on the nine cases with PBE and shear as the predictor variables and F_w and as the response variables. Table 3.3 lists the results of this analysis.

Table 3.3. Percent of variation (coefficient of determination or R^2) in tornadic intensity (F_w) that can be explained by the predictor variables PBE and shear for linear bivariate regression.

Case		R^2 FW
OK	(84117)	0.2852
WI-IL	(84118)	0.2581
IA	(84159)	0.5259
NE	(85130)	0.4266
OH-PA-NY	(85150)	0.1868
SD	(86209)	0.9766
KS	(86261)	0.1695
TX-LA	(87319)	0.0613
NC-VA	(88333)	0.0798

These results show a highly variable accounting for the variance in the data from case to case, even more so than UMAX and MDA did as predictors. Only 34% of the variance was accounted for when F_w was used as a response variable predicted by shear and PBE. However, as evident in the data, the coefficient of determination of shear and PBE to tornadic intensity did not even account for the accuracy that UMAX and MDA possessed in describing thunderstorm behavior. This could be due to many reasons, one of which is the temporal problem of having only two soundings available per day (excluding more timely soundings taken during special research experiments) versus the almost real-time analysis available with satellite imagery in determining UMAX and MDA. The other problem is spatial, that is, the sounding spacing network in existence today is not very refined, especially in the mesoscale region of analysis. This is particularly apparent when one radiosonde station is used for many developing cells, one of which may produce a tornado. However, for whatever the reason, we are led to a very important point: PBE and shear, derived exclusively from pre-storm data are not reliable generalized discriminators of tornadic behavior.

3.3 Statistical Analysis of F_w to UMAX, MDA, PBE and Shear

A "fine tuning" in the coefficient of determination of tornadic intensity to thunderstorm characteristics was done by using a quadvariate regression model with UMAX, MDA, PBE and shear as the predictor variables and F_w as the response variable (equation 2.5). Table 3.4 shows the results of quadvariate linear regression analysis

Table 3.4. Percent of variation (coefficient of determination or R^2) in tornadic intensity (F_w) that can be explained by the predictor variables UMAX, MDA, PBE and shear for linear quadvariate regression.

		R^2
Case		FW
OK	(84117)	0.4756
WI-IL	(84118)	0.9071
IA	(84159)	0.8671
NE	(85130)	0.9033
OH-PA-NY	(85150)	0.8636
SD	(86209)	0.9845
KS	(86261)	0.7574
TX-LA	(87319)	0.7230
NC-VA	(88333)	0.8030

on the nine outbreak cases. In contrast to the earlier bivariate results, the quadvariate linear regression was not as variable from case to case. Overall, F_w gave an estimate of tornadic intensity of 81%. In only one case, (OK), was the coefficient of determination drastically smaller than the other eight cases. This may be due to errors in the measurement of UMAX and MDA through the predominant use of poor resolution (4 km) IR satellite imagery over visible satellite imagery (1 km). Overall, three cases showed greater than 90% of the variance accounted for in the data, and six cases showed greater than 80% of the variance accounted for in the data.

Table 3.5 shows the overall comparison results of the two bivariate regression models and the quadvariate model. The apparent improvement in using the four variable model over the two variable model varied from case to case, from as little as 1% in the Wisconsin-Illinois case to as much as 15% in the Iowa outbreak. Overall the use of the quadvariate model over the bivariate regression model merely "fine-

Table 3.5. Percent of variation (coefficient of determination or R^2) in tornadic intensity (F_w) for linear regression of different regression schemes using UMAX, MDA, PBE and shear as the predictor variables.

		Coefficient of Determination (R^2)		
Case		UMAX/MDA	Shear/PBE	UMAX/MDA/Shear/PBE
OK	(84117)	0.4065	0.2852	0.4756
WI-IL	(84118)	0.8948	0.2581	0.9071
IA	(84159)	0.7214	0.5259	0.8671
NE	(85130)	0.8861	0.4266	0.9033
OH-PA-NY	(85150)	0.7658	0.1868	0.8636
SD	(86209)	0.7058	0.9766	0.9845
KS	(86261)	0.6326	0.1695	0.7574
TX-LA	(87319)	0.7193	0.0613	0.7230
NC-VA	(88333)	0.7724	0.0798	0.8030

tuned" the variation accounted for in the data.

3.4 Stratification of the Data

Up to now, the ability to relate the tornadic intensity of a given cell to a set of predictors depends on complete data. That is, once the breakpoint value of a given outbreak is known, then a reasonable hindcast as to that cell's tornadic intensity can be determined within that outbreak. However, as has been shown, the breakpoint varies from outbreak to outbreak (table 3.1). This does not lead to a very practical way of determining the possible tornadic intensities of thunderstorm cells if one does not know the breakpoint value beforehand, which makes forecasting them impossible.

Since each outbreak has a different breakpoint value and surface, the combined data was stratified based on each outbreak's d-value (see Table 3.1). Table 3.6 shows how the combined data set was stratified.

Table 3.6. D-value results of stratification of combined data set and their respective range within strata.

		d-Value
Stratified Case	d-Value Range	Difference
I (WI-IL; OH-PA-NY)	30.23 - 31.30	1.07
II (OK; IA)	17.00 - 19.21	2.21
III (SD; TX-LA)	20.52 - 21.47	0.95
IV (NE; KS; NC-VA)	12.21 - 13.40	1.19

Note that the range of d within a given case varied between 0.95 to 2.21, showing that the difference among a stratified group is small.

As was done with the individual outbreak cases, UMAX and MDA were plotted as a function of tornadic intensity within each stratified data set and all four exhibited a rising-ridge surface. Figures 3.23 -3.26

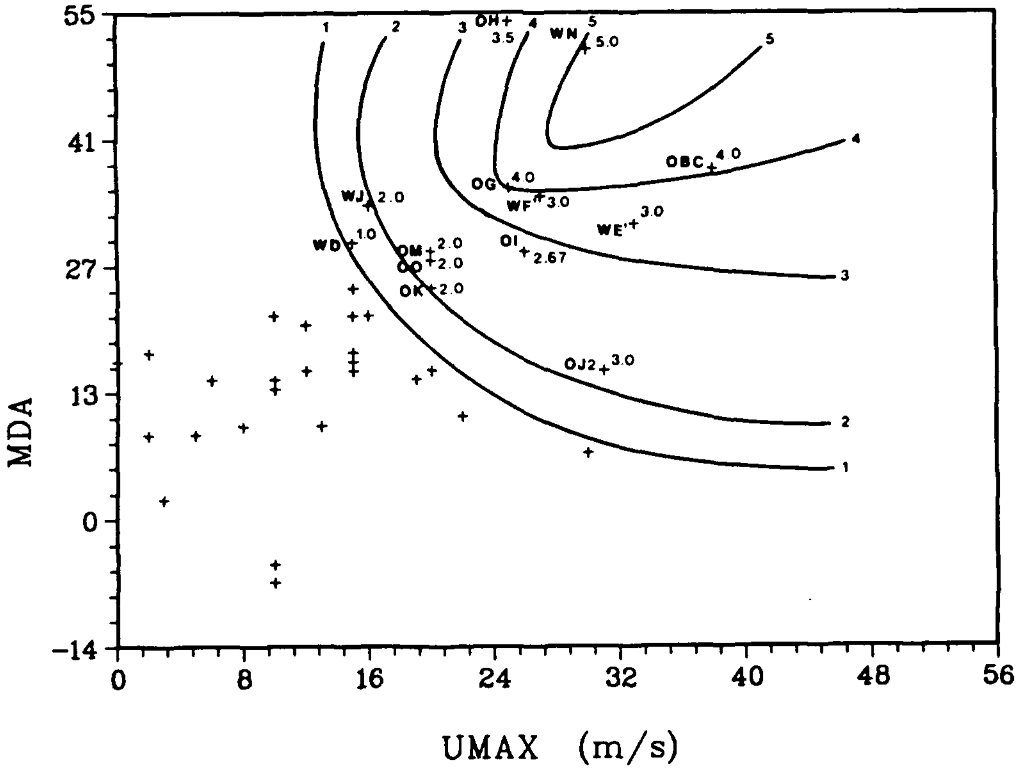


Figure 3.23. Rising-ridge plot analysis of UMAX, MDA versus F_W for the stratified data set I (OH-PA-NY; WI-IL). Letters indicate the cell of the outbreak, which are preceded by an O (OH-PA-NY) or a W (WI-IL). Values are the tornadic intensities of the associated cells (F_W). Cells without letters and values were non-tornadic.

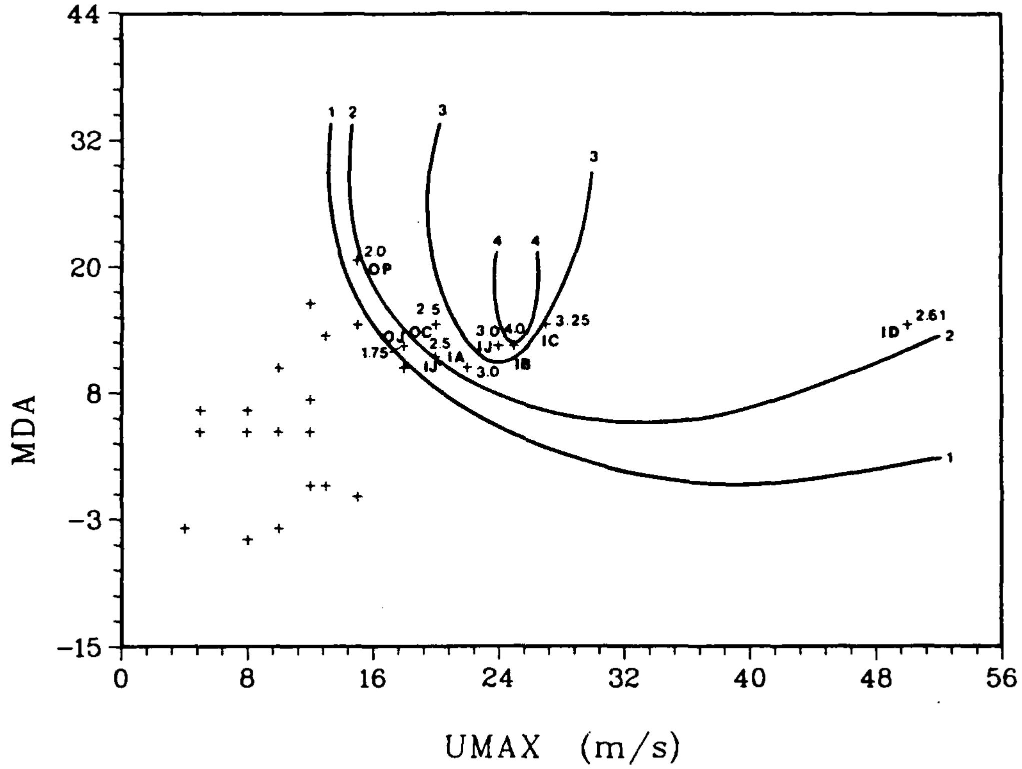


Figure 3.24. Rising-ridge plot analysis of UMAX, MDA versus F_W for the stratified data set II (OK; IA). Letters indicate the cell of the outbreak, which are preceded by an O (OK) or an I (IA). Values are the tornadic intensities of the associated cells (F_W). Cells without letters and values were non-tornadic.

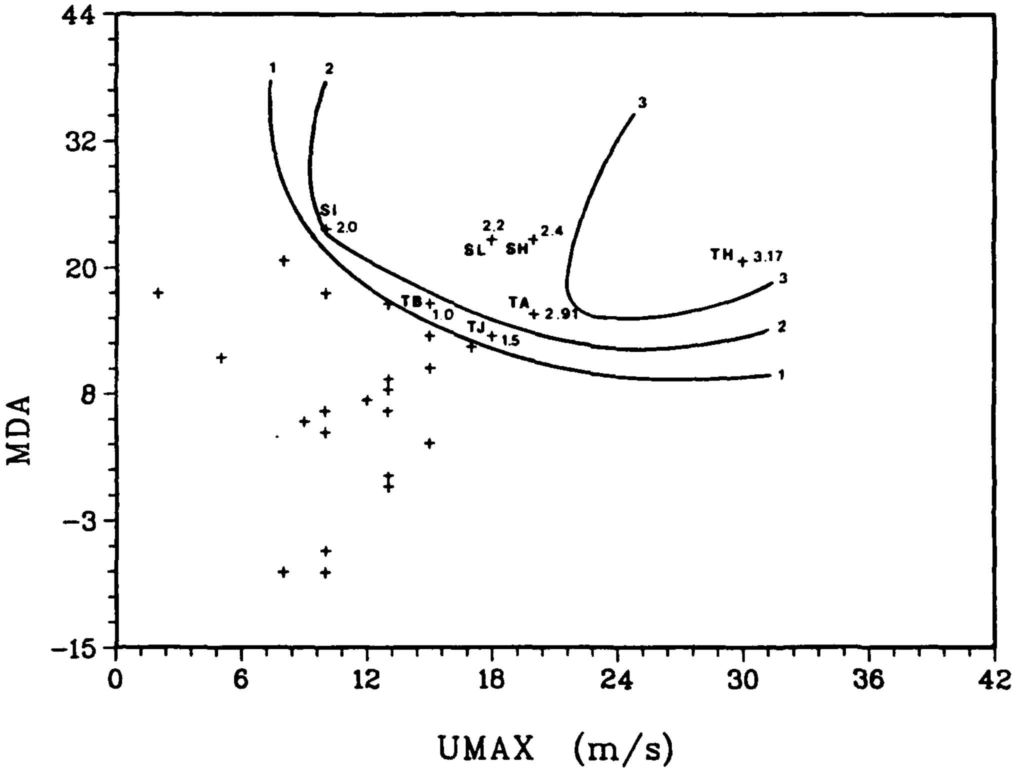


Figure 3.25. Rising-ridge plot analysis of UMAX, MDA versus F_w for the stratified data set III (SD; TX-LA). Letters indicate the cell of the outbreak, which are preceded by a S (SD) or a T (TX-LA). Values are the tornadic intensities of the associated cells (F_w). Cells without letters and values were non-tornadic.

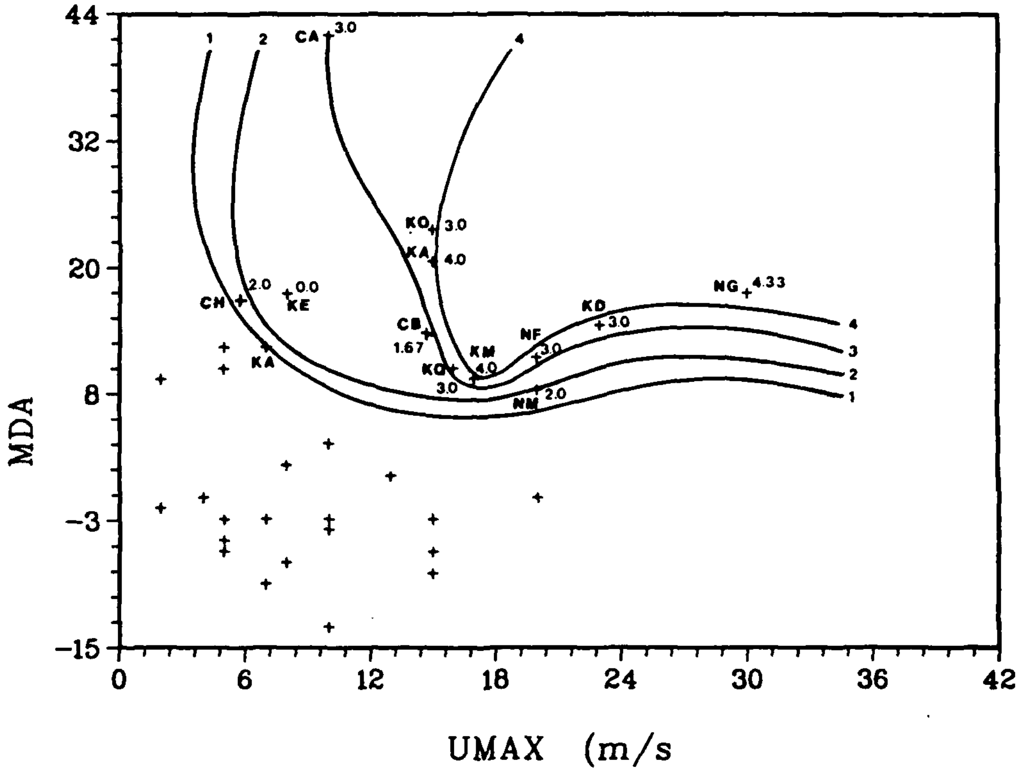


Figure 3.26. Rising-ridge plot analysis of UMAX, MDA versus F_W for the stratified data set IV (NE; KS; NC-VA). Letters indicate the cell of the outbreak, which are preceded by a N (NE), a K (KS) or a C (NC-VA). Values are the tornadic intensities of the associated cells (F_W). Cells without letters and values were non-tornadic.

depict this relationship. As can be seen from the plots, a fairly good fit of the data arrises due to this stratification. However, stratified case IV did not fit a normal, smooth surface as exhibited by the other cases. This may be due to analysis errors of UMAX and/or MDA.

In order to determine how well the slopes and intercepts of each outbreak surface matched each other, the outbreak's cells within each stratified data set were further classified as either tornadic or non-tornadic. That is, if a cell's F_w value was non-zero, then that cell was classified as tornadic (denoted as "1"). If the cell had a zero F_w then it was classified as non-tornadic (denoted as "0"). Further, outbreaks were given numbers within each stratified data set. That is, if there were two outbreaks in a given strata, then there were two numbers designating outbreak occurrence. Appendix 6.6 gives an explanation of the tests used for this statistical procedure. This type of classification was used solely to predict tornadic or non-tornadic occurrence, and not severity. Earlier it was shown that F_w was a better variable in denoting tornadic intensity than F_w^2. Similarly, a classification of tornadic or non-tornadic occurrence in the data will further limit the range of values for F_w. This test showed that the outbreak slope surfaces within the stratified data sets were not different from one another (table 3.7), and that the groups of tornadic and non-tornadic cells were different from case to case. By picking a p-value of 0.05 (the smallest significant level at which our results would lead to a rejection of the hypothesis that the variables examined are not needed (in this case any p-value $<$ 0.05),

Table 3.7. Type I sums of squares and p-values of UMAX, MDA, Outbreak and their interaction in determining whether outbreak intercepts were the same within a stratified data set.

Stratified Case I Variable	degrees of freedom	Type I Sums of Squares	p-value
UMAX	1	4.0212	0.0001
MDA	1	1.5810	0.0001
Outbreak	1	0.1704	0.1446
UMAX*Outbreak	1	0.0100	0.7195
MDA*Outbreak	1	0.3325	0.0447

Stratified Case II Variable	degrees of freedom	Type I Sums of Squares	p-value
UMAX	1	3.1958	0.0001
MDA	1	0.4267	0.0458
Outbreak	1	0.0248	0.6149
UMAX*Outbreak	1	0.0015	0.9001
MDA*Outbreak	1	0.3643	0.0632

Stratified Case III Variable	degrees of freedom	Type I Sums of Squares	p-value
UMAX	1	2.0389	0.0001
MDA	1	0.8485	0.0061
Outbreak	1	0.3045	0.0831
UMAX*Outbreak	1	0.0142	0.6984
MDA*Outbreak	1	0.0120	0.7225

Stratified Case IV Variable	degrees of freedom	Type I Sums of Squares	p-value
UMAX	1	1.7592	0.0001
MDA	1	1.8101	0.0001
Outbreak	2	0.2339	0.1835
UMAX*Outbreak	2	0.0522	0.6747
MDA*Outbreak	2	0.0528	0.6717

the results from table 3.7 show a strong difference in UMAX and MDA from outbreak to outbreak but not in outbreak intercepts (denoted as OUTBREAK in table 3.7). The interaction of outbreak upon UMAX and MDA (differences in slopes of UMAX and MDA) also shows little difference. This suggests that within a strata, outbreaks do not play a role in determining tornadic occurrence.

One can also look at the type I sums of squares to get an idea as to what may need to be included in the analysis. If the type I sums of squares is relatively high, then we would tend to keep that data. If it was low, then that data could be left out of the analysis. So, the type I sums of squares in table 3.7 shows that there are differences of UMAX, MDA among each stratified data set. This means that UMAX and MDA are good indicators of tornadic occurrence. However, outbreak intercepts among the stratified data sets are about the same, showing little difference in tornadic occurrence from outbreak to outbreak. Therefore, one would expect the data from one outbreak to overlay exactly on another case.

It should be noted that the above test was also done on tornadic severity (F_w) vice tornado occurrence for all the stratified cases. However, the results were not as obvious using F_w, since there may be some question as to whether or not the slopes may be different in the stratified cases.

By performing regression analysis on these four stratified groups of data, significant results appear. These are shown in table 3.8.

Table 3.8. Coefficients of determination from bivariate and quadvariate regressions of various orders for stratified data from combined data set using F_w as the response variable.

		R^2	
Case	Predictor	Linear	Quadratic
I	UMAX, MDA	0.7573	0.8626
	Shear, PBE	0.1596	0.2147
	UMAX, MDA, Shear, PBE	0.7984	0.9475
II	UMAX, MDA	0.5813	0.7976
	Shear, PBE	0.2047	0.4259
	UMAX, MDA, Shear, PBE	0.6641	0.9115

Table 3.8 (continued).

III	UMAX, MDA	0.6323	0.7850
	Shear, PBE	0.1981	0.3496
	UMAX, MDA, Shear, PBE	0.7343	0.9343
IV	UMAX, MDA	0.7206	0.8191
	Shear, PBE	0.1540	0.2919
	UMAX, MDA, Shear, PBE	0.7577	0.9263

The overall results from this table are summarized in table 3.9. The comparison of the linear regression analysis between individual and stratified data sets shows encouraging results. There was only a 5% difference in the overall percentage of variance explained by using UMAX and MDA as the predictors for F_w. Only a 7% difference ensued when PBE and shear was added to the bivariate model. No comparison could be done with the quadratic regression analysis since it could not be used for the individual outbreak data sets.

Table 3.9. Overall percentage of variance explained (R^2) of linear bivariate and quadvariate regressions between stratified data sets and individual outbreak data sets using F_w as the response variable.

	Average R^2	
Outbreak Analysis	UMAX, MDA	UMAX, MDA, PBE, Shear
Individual Cases	0.7227	0.8094
Stratified Cases	0.6729	0.7386
Difference	0.0498	0.0708

In contrast to the regression results of the stratified data sets, regression was also carried out on the combined data set to see how the two data grouping schemes would differ.

For example, when the outbreaks were grouped into one data set and examined under a regression scheme the results were not good. First and second order multivariate regression was done on the combined data set and it showed little significant results, as shown in table 3.10.

Table 3.10. Coefficients of determination (R^2) from bivariate and quadvariate regressions of various orders for combined data from all cases using F_w as the response variable.

	R^2	
Predictor	Linear	Quadratic
UMAX, MDA	0.5792	0.6003
Shear, PBE	0.0120	0.0605
UMAX, MDA, Shear, PBE	0.5809	0.6322

Even the addition of PBE and shear in the quadvariate model did little to improve the R^2 values. These poor results may be explained by the fact that each outbreak took place under different synoptic weather conditions. Furthermore, these poor results imply the non-conformity of the individual rising-ridge surfaces when overlaid.

In contrast to the test of surface slopes among individual statified data sets, a similar test was performed on the one combined data set. Table 3.11 shows the results. Again picking a p-value of 0.05 as significant, the results show that there are differences in

Table 3.11. Type I (sequential) sums of squares and p-values of UMAX, MDA, Outbreak and their interaction in determining whether outbreak intercepts were the same within the combined data set.

Variable	degrees of freedom	Type I Sums of Squares	p-value
UMAX	1	11.9258	0.0001
MDA	1	3.0655	0.0001
Outbreak	8	3.0132	0.0001
UMAX*Outbreak	8	0.7257	0.3470
MDA*Outbreak	8	0.7801	0.2952

outbreak intercepts among the data, as well as a strong dependance on UMAX and MDA as predictors of tornadic occurrence. This is substantiated by the relatively high Type I sums of squares for UMAX, MDA and OUTBREAK. Therefore, one would not necessarily expect the data from one outbreak to overlay exactly on another case. These differences in surfaces explains the relatively poor regression results in the combined data set.

In an attempt to bring all nine cases into some type of common form, the data was non-dimensionalized by case breakpoint value. Table 3.12 shows the results of this procedure. As can be seen from this and table 3.10, regression results of the non-dimensionalized data improved somewhat over those of the combined data set, but not enough as compared to the results of individual cases.

Table 3.12. Coefficients of determination from bivariate and quadvariate regressions of various orders for non-dimensionalized combined data using F_w as the response variable.

	R^2	
Predictor	Linear	Quadratic
UMAX, MDA	0.6412	0.7147
Shear, PBE	0.0120	0.0605
UMAX, MDA, Shear, PBE	0.6437	0.7514

As can be seen from the analysis of stratified data sets and the combined and non-dimensionalized data sets, the stratification of the data yields good results when compared to the individual cases. This was especially true when examining the results from the combined and non-dimensionalized data sets to individual outbreaks. Therefore, this suggests that within these stratified groups, some similar type of meteorological phenomena may be occurring in order to bring the response surfaces closer together (similar slopes and intercepts) within the strata.

4. CONCLUSIONS

The basic goal of this study was to eliminate the two-step process that Schrab (1988) and Anderson and Schrab (1988) described in relating UMAX and MDA in a bivariate regression model to predict tornadic intensities of thunderstorms. Thunderstorm anvil behavior parameterized by UMAX (the downstream mass flux of an anvil outflow plume) and MDA (the rightward deviation angle of the anvil from the storm-relative flow) showed a relationship on a plotted surface that distinquished tornadic from non-tornadic cells. The use of a four variable (quadvariate) regression model with UMAX, MDA, PBE and shear as the predictor variables was not sufficient to predict the tornadic intensity of a given observed cell, regardless of its outbreak family. Therefore, the elimination of the two-step process did not occur.

The first conclusion was that the best variable for denoting tornadic intensity was the weighted mean value (F_w). Better R^2 values resulted when F_w was used rather than a weighted mean square system of measurement, called F_w^2. This was probably due to the data distribution occurring in the extreme tails of the distribution curve for F_w^2. This was not the case in using F_w as a measure of tornadic intensity.

Another conclusion was a fine-tuning of the analysis results occurred with the use of the quadvariate model, due to the addition of PBE and low-level vertical wind shear. The amount of fine tuning varied from 1% to as much as 15%. Attempts to combine the data into one lumped group failed to improve the coefficient of determination. It fell to just over 58% with the four variable model. Non-dimensionalization of this combined data set only improved R^2 slightly to 0.64. Only linear regression could be used due to the limited amount of observations available per outbreak.

The other conclusion was that the data could be combined in such a way as to improve the coefficient of determination. This was done by stratifying the data sets based on outbreak breakpoint value. Specifically, the distance of each individual outbreak's breakpoint value from the origin was computed. This led to the grouping into four stratified data sets. The stratification of combined data set yielded promising results. Linear regression gave encouraging results, especially when compared to the individual regression analysis. The bivariate predictors UMAX and MDA on stratified groups gave an average correlation of 67%, whereas the quadvariate model accounted for 74%. This came within 5% of the individual case bivariate regression results, and within 7% for the quadvariate regression analysis. This seemed to validate the idea that the stratified data sets best describes a certain outbreak situation. These situations may be different types of meteorological parameters such as mesoscale lows, mid-level disturbances, and squall line formation.

Quadratic regression could not be carried out on the stratified cases since the number of observations per strata were insufficient to overcome the problem of overfitting the data to the models. For bivariate regression analysis we need 40 observations to justify using quadratic regression. The quadvariate model will require at least 100 observations to justify the use of quadratic regression analysis.

5. FUTURE RESEARCH

Data stratification of the nine outbreak studies yielded promising results, especially when compared to the regression results of the combined data set. There seems to be a good relationship among outbreaks within strata, suggesting that other factors may be common to outbreak cases within each stratified data set. However, the breakpoint value of a given outbreak has to be found in order to classify it into one of the stratified data sets. As a consequence, the two-step process is still needed to predict tornadic intensities of observed thunderstorm cells.

In order to implement some sort of operational forecast to determine which observed cell will produce a tornado, further research must be done to try to find a way to predict the outbreak's breakpoint value. Some sort of analysis of these stratified groups is in order to relate what we now have to other known meteorological parameters. Work by Maddox and Doswell (1983) and Miller (1972) suggest that other meteorological phenomena may be present to explain the unique triggering mechanisms that occur from case to case. Table 5.1 lists some of these parameters.

Table 5.1. Summary of key parameters (after Miller, 1972)

Rank	Parameter	Weak (W)	Moderate (M)	Strong (S)
1	500 mb vorticity advection	Neutral or NVA	PVA-Contours Cross Vorticity Pattern $\le 30^\circ$	PVA-Contours Cross Vorticity Pattern $> 30^\circ$
2	Stability-Totals Index	$TI \le 50$	$50 < TI \le 55$	$TI > 55$

Table 5.1. (continued)

Rank	Parameter	Weak (W)	Moderate (M)	Strong (S)
3	500 mb wind speed	WS ≤ 35 kt	35 < WS ≤ 50 kt	WS > 50 kt
4	300-200 mb wind speed (upper-level max)	WS ≤ 55 kt	55 < WS ≤ 85 kt	WS > 85 kt
5	850 mb wind speed	WS ≤ 20 kt	20 < WS < 35 kt	WS ≥ 35 kt
6	850 mb dewpoint	Td ≤ 8°C	8 < Td ≤ 12°C	Td > 12°C
7	850 mb temp ridge location	East of moist axis	Over moist axis	West of moist axis
8	700 mb temp no. change line(12-h)	Winds cross line ≤ 20°	Winds cross line>20 &≤40°	Winds cross line > 40°
9	700 mb dry intrusion	N/A or weak 700 mb winds	Winds from dry to moist intrude at angle of < 40° and are≥15 kt	Winds intrude at an angle of ≥ 40° and are≥25 kt
10	12-h surface pressure fall	< 1 mb	1 to 5 mb	> 5 mb
11	500 mb height change (12-h)	< 30 m	≥ 30 m & ≤ 60 m	> 60 m
12	Surface pressure over threat area	≥ 1010 mb	< 1010 mb and ≥ 1005 mb	< 1005 mb
13	Surface dewpoint	Td ≤ 55°F	55 < Td < 65°F	Td ≥ 65°F

Continuation of this research should include determining some of the parameters present in each outbreak (from table 5.1) by attempting to relate these parameters in some way (either by regression analysis or perhaps some sort of discriminant analysis) to each stratified data case. This may lead to some type of forecastable breakpoint value to relate observed cells to an already dtermined stratified surface. If some sort of common phenomena occur within a stratified data set, this may lead to a possible solution to this problem. Such criteria should focus upon the important physical mechanisms leading to the development of strong storms within an environment. The importance of favorable surface patterns and pronounced thermal boundaries, mesoscale features such as vertical motion forced by lower-tropospheric warm air advection need to be further investigated.

Appendix 6.1

Distribution of Tornadoes, Intensities and Injuries

This appendix breaks out the distribution of tornadoes and their intensities associated with each thunderstorm cell at the observed time of development. Along with each cell's tornadic intensities, a list of deaths and injuries with that cell is included. All times given are Greenwich Mean Time (GMT).

Table 6.1.1. Oklahoma Outbreak of 26 April 1984 (84117)

Cell	Time	F_M	F_M^2	Distribution	Killed	Injured
A	1930-2030	0.00	0.00			
B	1930-2030	0.00	0.00			
C	2014-2044	2.50	6.50	1F1,1F2	0	5
D	2014-2044	0.00	0.00			
E	2014-2044	0.00	0.00			
F	2044-2130	0.00	0.00			
G	2044-2130	0.00	0.00			
H	2200-2230	0.00	0.00			
I	2200-2300	0.00	0.00			
J	2200-2300	1.25	1.75	3F0,1F1	0	0
K	2200-2300	0.00	0.00			
L	2200-2300	0.00	0.00			
M	2230-2300	0.00	0.00			
N	2230-2300	0.00	0.00			
O	2230-2300	0.00	0.00			
P	2053-2130	2.00	4.00	1F1	0	0

Table 6.1.2. Wisconsin-Illinois Outbreak of 27 April 1984 (84118)

Cell	Time	F_M	F_M^2	Distribution	Killed	Injured
A	1715-1800	0.00	0.00			
B	2000-2100	0.00	0.00			
C	2030-2100	0.00	0.00			
D	2100-2200	0.00	0.00			
D'	2230-2330	1.00	1.00	1F0	0	0
E	2130-2230	0.00	0.00			
E'	2230-2330	3.00	10.00	1F1,1F3	1	5
F	2130-2230	0.00	0.00			
F'	2230-2300	3.00	13.00	1F0,1F4	1	14
G	2130-2200	0.00	0.00			
H	2145-2245	0.00	0.00			
J	2230-2300	2.00	4.00	1F1	0	0
N	2100-2200	5.00	25.00	1F4	1	19
Q	2100-2130	0.00	0.00			
S	2215-2245	0.00	0.00			
T	1600-1700	0.00	0.00			

Table 6.1.3. Iowa Outbreak of 7 June 1984 (84159)

Cell	Time	F_M	F_M^2	Distribution	Killed	Injured
A	1931-2030	3.00	9.00	1F2	0	8
B	2000-2100	4.00	16.00	1F3	0	3
C	2000-2100	3.25	11.75	1F1,2F2,1F4	3	70
D	1901-2000	2.61	7.52	1F0,8F1,12F2,2F3	0	32
E	2200-2230	0.00	0.00			
E'	2230-2330	0.00	0.00			
G	1801-1901	0.00	0.00			
H	1801-1901	0.00	0.00			
I	1701-1801	0.00	0.00			
J	2245-2330	3.00	9.00	3F2	0	1
J'	2245-2330	2.50	6.50	1F1,1F2	1	3
L	1901-2000	0.00	0.00			

Table 6.1.4. Nebraska Outbreak of 10 May 1985 (85130)

Cell	Time	F_w	F_w^2	Distribution	Killed	Injured
A	2000-2100	4.00	16.00	2F3	0	0
B	2000-2100	0.00	0.00			
C	2000-2100	0.00	0.00			
D	2100-2200	0.00	0.00			
E	2130-2230	0.00	0.00			
F	2144-2244	3.00	10.00	1F1,1F3	0	2
G	2200-2300	4.33	19.67	1F2,2F4	0	2
H	2130-2230	0.00	0.00			
I	2200-2300	0.00	0.00			
J	2230-2330	0.00	0.00			
K	2314-2344	0.00	0.00			
L	2214-2314	0.00	0.00			
M	2314-2344	2.00	4.00	1F1	0	0

Table 6.1.5. Ohio-Pennsylvania-New York Outbreak of 31 May 1985 (85150)

Cell	Time	F_w	F_w^2	Distribution	Killed	Injured
A	2000-2100	0.00	0.00			
B(C)	2000-2100	4.00	17.14	2F1,2F2,4F3,6F4	44	337
D	2000-2100	0.00	0.00			
E	2030-2100	0.00	0.00			
F	2030-2100	0.00	0.00			
G	2130-2230	4.00	16.67	1F2,1F3,1F4	4	63
H	2130-2230	3.50	17.42	1F0,1F2,1F5	18	32
I	2200-2300	2.67	8.67	1F0,1F2,1F3	9	140
J2	2300-2344	3.00	10.00	1F1,1F3	1	25
K	2230-2330	2.00	4.00	1F1	0	0
L	2230-2330	0.00	0.00			
M	2300-2344	2.00	4.00	1F1	0	0
O	2300-2344	2.00	4.00	1F1	0	0
Q	2030-2100	0.00	0.00			
R	2300-2344	0.00	0.00			
U	2030-2100	0.00	0.00			
V	2000-2100	0.00	0.00			
W	2030-2130	0.00	0.00			
X	2130-2230	0.00	0.00			
Y	2130-2230	0.00	0.00			
Z	2130-2230	0.00	0.00			
AA	2144-2244	0.00	0.00			

Table 6.1.6. South Dakota Outbreak of 28 July 1986 (86209)

Cell	Time	F_M	F_M^2	Distribution	Killed	Injured
A	1714-1800	0.00	0.00			
B	1830-1900	0.00	0.00			
C	1900-2000	0.00	0.00			
D	1900-2000	0.00	0.00			
E	1914-2000	0.00	0.00			
F	1914-2000	0.00	0.00			
G	1930-2000	0.00	0.00			
H	1930-2000	2.40	6.40	1F0,1F1,3F2	0	0
I	2000-2100	2.00	4.50	1F0,2F1,1F2	0	0
J	2130-2230	0.00	0.00			
K	2130-2230	0.00	0.00			
L	2200-2230	2.20	7.40	3F0,1F2,1F4	0	1

Table 6.1.7. Kansas Outbreak of 18 September 1986 (86261)

Cell	Time	F_M	F_M^2	Distribution	Killed	Injured
A	2001-2101	0.00	0.00			
B	2001-2101	0.00	0.00			
C	2001-2101	0.00	0.00			
D	2101-2201	3.00	9.00	1F2	0	0
E	2131-2231	0.00	0.00			
G	2131-2231	0.00	0.00			
H	2131-2231	0.00	0.00			
I	2131-2231	0.00	0.00			
J	2201-2301	0.00	0.00			
K	2231-2301	0.00	0.00			
L	0001-0101	0.00	0.00			
M	0001-0101	4.00	16.00	1F3	0	7
N	0001-0101	0.00	0.00			
O	0001-0101	3.00	9.00	1F2	0	0
P	0001-0101	0.00	0.00			
Q	0101-0201	3.00	9.00	2F2	0	0

Table 6.1.8. Texas-Louisiana Outbreak
of 15 November 1987 (87319)

Cell	Time	F_M	F_M^2	Distribution	Killed	Injured
A	1531-1631	2.91	9.27	1F0,2F1,5F2,3F3	10	171
B	1531-1631	1.00	1.00	1F0	0	0
C	1601-1701	0.00	0.00			
D	1601-1701	0.00	0.00			
E	1631-1731	0.00	0.00			
F	1916-2016	0.00	0.00			
G	1831-1901	0.00	0.00			
H	2201-2301	3.17	15.75	1F0,1F1,1F5	1	116
I	2101-2201	0.00	0.00			
J	2231-2331	1.50	2.50	1F0,1F1	0	0
K	2201-2231	0.00	0.00			
L	2046-2201	0.00	0.00			
M	1946-2016	0.00	0.00			
N	2201-2301	0.00	0.00			
O	2316-2346	0.00	0.00			
P	0031-0131	0.00	0.00			

Table 6.1.9. North Carolina-Virginia Outbreak
of 28 November 1988 (88333)

Cell	Time	F_M	F_M^2	Distribution	Killed	Injured
A	0501-0601	3.00	11.67	1F0,1F2,1F4	4	154
B	0731-0831	1.67	3.00	1F0,2F1	0	3
C	0001-0101	0.00	0.00			
D	0001-0101	0.00	0.00			
E	0001-0101	0.00	0.00			
F	0031-0131	0.00	0.00			
G	0131-0231	0.00	0.00			
H	0431-0531	2.00	4.00	1F1	0	0
I	0601-0701	0.00	0.00			
J	0531-0631	0.00	0.00			
K	0531-0631	0.00	0.00			
L	0631-0731	0.00	0.00			
M	0501-0601	0.00	0.00			

Appendix 6.2

Fujita Tornado Scale Rating System

The Fujita-scale (F-scale) wind speed classification system was designed to assess tornado wind speeds in a standardized rating scheme. The following are F-scale damage specifications in relating the strength of a tornado to the apparent damage associated with its wind speed as represented by Fujita (1981):

(F0) Gale Tornado 18-32 m s^{-1} (40-72 mph): Light damage. Some damage to chimneys; break branches off trees; push over shallow-rooted trees; damage sign boards.

(F1) Moderate Tornado 33-49 m s^{-1} (73-112 mph): Moderate damage. The lower limit (73 mph) is the beginning of hurricane wind speed; peel surface off roofs; mobile homes pushed off foundations or overturned; moving autos pushed off the roads.

(F2) Significant Tornado 50-69 m s-^{1} (113-157 mph): Considerable damage. Roofs torn off frame houses; mobile homes demolished; boxcars pushed over; large trees snapped or uprooted; light-object missiles generated.

(F3) Severe Tornado 70-92 m s^{-1} (158-206 mph): Severe damage. Roofs and some walls torn off well-constructed houses; trains overturned; most trees in forest uprooted; heavy cars lifted off ground and thrown.

(F4) Devastating Tornado 93-116 m s^{-1} (207-260 mph): Devastating damage. Well-constructed houses leveled; structure with weak foundation blown off some distance; cars thrown and large missiles generated.

(F5) Incredible Tornado 117-142 m s^{-1} (261-318 mph): Incredible damage. Strong frame houses lifted off foundations and carried considerable distance to disintegrate; automobile-sized missiles fly through the air in excess of 100 m; trees debarked; incredible phenomena occur.

(F6-F12) 142 m s^{-1} to Mach 1, the speed of sound. The maximum wind speeds of tornadoes are not expected to reach the F6 wind speeds.

Appendix 6.3

Low-Level Vertical Wind Shear Program Listing

```
Program Shear;
{
   This program computes vertical wind shear in several forms:
      - Low-level vertical wind shear (0-4km) --
            Either as an integrated sum of 200m levels, or as a vector
            difference between the surface and 4000m.  The integral of
            the shear vector magnitudes will indicate the presence of
            "looping" in the sounding profile, indicative of low-level
            jet influences.
      - User-specified levels of vertical wind shear --
            Either as an integerated sum of 200m levels or as a vector
            difference between the two specified levels.
        These previous two are by input data file as created by McIDAS
            in format as specified in INFO procedure below.
      - User-input winds and levels via keyboard interface

   This program will also plot a raob sounding profile from the McIDAS
      data input file on screen.

   This program will also convert the input data file from McIDAS
      format to a format used by the PBE program on MEAVAX, ready to
      download and use to calculate PBE.

   The program interpolates the input sounding data (T,Td,P,winds,Z)
      approximately every 200m (50 mb) and integrates it over the
      total height dz.  This method of computing shear was based on
      the techniques used by Blechman in 1979 and also by Rasmussen &
      Wilhelmson in 1983.

   This program was written in Turbo Pascal 4.0, implemented on an
      IBM PC/XT/AT with EGA capability.

   This program contains the following external files, and must be
     included on the same disk drive and/or directory in order to run:
          Turbo Pascal 4.0 Drivers:
                CRT.DOC         ( CRT interface driver )
                GRAPH.TPU       ( Graphics drivers     )
                PRINTER.DOC     ( Printer driver       )
                EGAVGA.BGI      ( EGA graphics drivers )

          Shear Program Externals:
                CONVMCID.AS     ( PBE data conversion procedure    )
                PLOT.ROB        ( Raob sounding plotting procdure )

    Dale R. Perry
    North Carolina State University     12/14/88
}
```

```
uses
   Graph,                     { Borland Graphics Unit }
   Printer,                   { Borland Printer  Unit }
   CRT;                       { Borland CRT BIOS Unit }

const
   MaxRaob    = 100;          { Maximal entries in raob array }
   ExtMaxRaob = 101;

type
   RaobData = record          { Raob data structure }
                 P_mb,T_C,TD_C,
                 Dir,Spd,
                 Z_m,Theta_K,W: Real
              end;
   RaobRange = 1..MaxRaob;
   DataArray = array[RaobRange] of RaobData;
   StnString = string[5];
   DummyStr  = string[6];

var
   Raob                : DataArray;
   StnID, Date, ZTime : StnString;  { Identifier & date of raob file }
   Lat, Lon,                        { Vars in McIDAS input files:    }
   DewPt, PotTemp,                  {               "                }
   ConvTemp,                        {               "                }
   EqvPTemp, MaxT,                  {               "                }
   MixRatio,                        {               "                }
   PrecipWtr, EqvP,                 {               "                }
   LIndex, Kindex,                  {               "                }
   TT, Sweat           : Real;      {               "                }
   Answer              : Char;      { Yes or no user prompt          }
   IOCode,                          { Var to contain IO err test code}
   EndofData,                       { Index last data point in file  }
   TestifFirstRun,                  { First time for running program }
   Choice,                          { Program function choice        }
   Code,                            { Error code string functions    }
   LCV                 : Integer;   { Loop control variable          }
   FileName            : string[14]; { Name of data file             }
   AlreadyReadInData,               { Test for existing data file    }
   GoodInput,                       { Check for valid input          }
   Over                : Boolean;   { Value for program termination  }

{ ****************************************************************** }

   procedure ProgramInfo;

   begin    { ProgramInfo }
     ClrScr;
     TextColor(Cyan);
```

```
writeln; writeln;
writeln('  This program will read in upper air data [either from'
        ,' McIDAS or by keyboard]');
writeln('  and processes that data (wind shear and/or PBE) based'
        ,' upon your input.');
writeln;
writeln('     Ways to enter the data:');
writeln;
writeln('        - You can use an input file in the format of ',
        'McIDAS output;');
writeln('        - You can input the data directly via keyboard '
        ,'by program prompts;');
writeln('        - You can combine input by both keyboard entry '
        ,'and McIDAS input file.');
writeln;
writeln('  The format of the input file MUST be by the following'
        ,' McIDAS commands');
writeln('  issued in sequence (this creates the data file at ',
        'Madison in the proper ');
writeln('  format for program execution and brings down the LW ',
        'file data to the PC).');
writeln('  This will be the input data file used by this ',
        'program:');
writeln;
writeln('       (1) .UL LIST station time DAY=yyddd MDF=xxxx ',
        'PTEMP=YES DEV=F filename');
writeln('       (2) .SENLW filename filename TRANS=A DRIVE=A');
writeln;
write(' Hit any key to continue...');
repeat until Keypressed;
  Answer:= ReadKey;
ClrScr;
writeln; writeln;
writeln('     The program will then ask for various inputs:');
writeln;
writeln('        - Prompt for user input or input file data ',
        '[& filename of input data]');
writeln;
writeln('        - If user input, than prompt for winds & height'
        ,' (in meters) of that');
writeln('          level [program will prompt twice, once for ',
        'each level]');
writeln;
writeln('        - If input file, than prompts for the levels at'
        ,' which shear is to be');
writeln('          computed');
writeln;
writeln('           -- Program computes the height based in data'
        ,' in input file, and will');
writeln('              find the closest values you requested in',
        ' the data.');
```

```
     writeln;
     writeln('        - Program plots an upper air sounding.');
     writeln;
     writeln('        - Program also converts the McIDAS data files ',
             'to a format used');
     writeln('            by the Potential Buoyant Energy (PBE) ',
             'program (on MEAVAX).');
     writeln;
     TextColor(141);   { Blinking LightMagenta }
     write(' ':25,'Hit any key to start the program...');
     repeat until Keypressed;
       Answer:= readkey;
     TextColor(Cyan)
   end;     { ProgramInfo }

{ ******************************************************************** }

   procedure BoxOutline;
   {
       Writes the menu and sub-menu boxes on the CRT
   }
   var
     Box: Integer;

   begin    { BoxOutline }
     ClrScr;
     TextColor(LightBlue);
     writeln;
     writeln; writeln; writeln;
     write(' ',#$C9);
     for Box:= 1 to 76 do
       write(#$CD);
     writeln(#$BB);
     for Box:= 1 to 12 do
       writeln(' ',#$BA,' ':76,#$BA);
     write(' ',#$C8);
     for Box:= 1 to 76 do
       write(#$CD);
     writeln(#$BC)
   end;     { BoxOutline }

{ ******************************************************************** }

   procedure Initialize;
   {
      Initializes the raob data structure to missing values '999.0'.
      These are the variables that appear in the McIDAS data file
      accessed by this program, specified in the format given by the
      'ProgramInfo' procedure.
   }
   begin    { Initialize }
```

```
     for LCV:= 1 to MaxRaob do
       begin
          Raob[LCV].P_mb    := 999.0;
          Raob[LCV].T_C     := 999.0;
          Raob[LCV].TD_C    := 999.0;
          Raob[LCV].Dir     := 999.0;
          Raob[LCV].Spd     := 999.0;
          Raob[LCV].Z_m     := 999.0;
          Raob[LCV].Theta_K := 999.0;
          Raob[LCV].W       := 999.0
       end
   end;     { Initialize }

{ ****************************************************************** }

   procedure CheckMissingData(DummyC: DummyStr;
                              var Missing: Boolean);
   {
      Checks for missing data fields, denoted as blanks in the file
   }
   const
      MissingData = '     ';

   begin    { CheckMissingData }
      if DummyC = MissingData then
         Missing:= true
      else
         Missing:= false
   end;     { CheckMissingData }

{ ****************************************************************** }

   procedure InsertZero(DummyA: DummyStr;
                        var Result: Real);
   {
      Checks for error in data format..i.e., if no zero exists
      before decimal place, it then inserts one.  Also checks for
      a minus sign before the decimal, again inserting a zero
      if found.
   }
   const
     MinusDec = '-.';  { Negative & missing leading zero          }
     BlankDec = ' .';  { Blank & missing leading zero             }
     Zero = '0';

   var
     NoMinus,          { Position in data entry where no minus is }
     NoZero: Integer;  { Position in data entry where no zero is  }
     DummyB: DummyStr;

   begin    { InsertZero }
```

```
  DummyB:= DummyA;
  NoZero:= pos(BlankDec,DummyB);
  NoMinus:= pos(MinusDec,DummyB);
  if NoZero = 0 then
    if NoMinus = 0 then   { Number is okay..no trailing zeroes }
      val(DummyB,Result,Code)
    else
      begin      { Insert a '0' between minus sign and decimal }
        Delete(DummyB,NoMinus-1,1);
        Insert(Zero,DummyB,NoMinus);
        val(DummyB,Result,Code)
      end
  else
    begin        { Insert a '0' before the decimal place }
      Delete(DummyB,NoZero,1);
      Insert(Zero,DummyB,NoZero);
      val(DummyB,Result,Code)
   end
end;     { InsertZero }

{ ***************************************************************** }

procedure ReadInData;
{
    Reads in the data from the McIDAS data file.  Procedure deter-
    mines where the data ends in the user-defined input file and
    fills the raob data structure with the pertinent info.
    Procedure also reads in the pre-determined met parameters (last
    three lines of input file) for possible later use [this version
    of the program doesn't make use of them].
}
const
   Keyword100 = ' 100.0'; {       End of data up to 100 mb        }
   Keyword    = 'PARCEL'; { Denotes start of pre-determined paras }

var
   Line      : string[80];    { Single read-in line from data file }
   Dummy1    : string[4];     { Dummy variables for individual     }
   Dummy2    : string[5];     {   data entries in line             }
   Check,                     { Checks for word match in data      }
   Dummy3    : DummyStr;      { Dummy variables for individual     }
   Dummy4    : string[7];     {   data entries in line             }
   DumVar    : LongInt;       { Dummy variable for Station ID      }
   Time,                      { Dummy time variable for conversion }

   CountLine : Integer;       { Line number for data file entries  }
   Result    : Real;          { Corrected data entry for program   }
   Found,                     { Boolean for end c  data field      }
   Missing   : Boolean;       { Boolean for mis..ng leading zeroes }
   Datafile  : text;          { Input file name                    }
```

```
begin   { ReadInData }
   TextColor(LightRed);
   {
       Test whether the file exists and
       display message if file doesn't
   }
   {$I-}
   repeat
     gotoxy(5,10);
     write('What filename does the data belong?:                ');
     gotoxy(42,10);
     readln(Filename);
     gotoxy(7,11); write(' ':70);
     assign(Datafile,Filename);
     reset(Datafile);
     IOCode:= IOResult;
     if IOCode <> 0 then
       begin
         gotoxy(7,11);
         write(^G,'ERROR!  File ',Filename,' does not exist.',
               ' Please re-enter...')
       end
   until IOCode = 0;
   {$I+}
   {
        Read in first line of data & determine where the data ends
   }
   CountLine:= 0;  { Initial data line # in file (excl. heading) }
   Found:= false;
   readln(DataFile,Line);   { Extract Station ID, Time(Z) }
   StnID:= copy(Line,11,5);
   Dummy3:= copy(Line,34,6);
   val(Dummy3,Dumvar,Code);
   Time:= Dumvar div 100;
   case Time of
           0 : ZTime:= '0000';
         600 : ZTime:= '0600';
        1200 : ZTime:= '1200'
      end;
   gotoxy(5,12);
   write('Reading in ',StnId,' ',ZTime,'Z data');
   {
      Extract remaining pertinent information
   }
   Date:= copy(Line,28,5);
   Dummy3:= copy(Line,52,6);
     val(Dummy3,DumVar,Code);
     Lat:= (DumVar div 100)/100;
   Dummy3:= copy(Line,60,6);
     val(Dummy3,DumVar,Code);
     Lon:= (DumVar div 100)/100;
```

```
  {
     Read in data up to 100 mb or when data ends
}
CountLine:= 0;
Found:= false;
repeat
  CountLine:= CountLine + 1;
  readln(Datafile,Line);
  Check:= copy(Line,2,6);
  if Check <> Keyword then
    begin
      Dummy3:= Check;
      val(Dummy3,Raob[CountLine].P_mb,Code);
      Dummy3:= copy(Line,11,5);
      InsertZero(Dummy3,Result);
      Raob[CountLine].T_C:= Result;
      Dummy3:= copy(Line,19,5);
      CheckMissingData(Dummy3,Missing);
      if not Missing then
      begin
        InsertZero(Dummy3,Result);
        Raob[CountLine].TD_C:= Result
      end;
      Dummy3:= copy(Line,27,5);
      CheckMissingData(Dummy3,Missing);
      if not Missing then
        val(Dummy3,Raob[CountLine].Dir,Code);
      Dummy3:= copy(Line,35,5);
      CheckMissingData(Dummy3,Missing);
      if not Missing then
        val(Dummy3,Raob[CountLine].Spd,Code);
      Dummy4:= copy(Line,42,7);
      val(Dummy4,Raob[CountLine].Z_m,Code);
      Dummy2:= copy(Line,52,5);
      val(Dummy2,Raob[CountLine].Theta_K,Code);
      Dummy3:= copy(Line,59,6);
      InsertZero(Dummy3,Result);
      Raob[CountLine].W:= Result
    end;
  if (Raob[CountLine].P_mb < 100.0) or (Check = Keyword) then
    Found:= true
until Found;
{
    Number of data entries in data file
}
EndofData:= Countline-1;
{
    Read in predetermined parameters
     (last three lines of data file)
}
readln(Datafile,Line);
```

```
        Dummy3:= copy(Line,18,6);  val(Dummy3,DewPt,Code);
        Dummy3:= copy(Line,35,6);  val(Dummy3,PotTemp,Code);
        Dummy3:= copy(Line,57,6);  val(Dummy3,EqvPTemp,Code);
        Dummy1:= copy(Line,69,4);  val(Dummy1,MixRatio,Code);
      readln(Datafile,Line);
        Dummy2:= copy(Line,15,5);  val(Dummy2,PrecipWtr,Code);
        Dummy2:= copy(Line,31,5);  val(Dummy2,ConvTemp,Code);
        Dummy2:= copy(Line,46,5);  val(Dummy2,MaxT,Code);
        Dummy2:= copy(Line,65,5);  val(Dummy2,LIndex,Code);
      readln(Datafile,Line);
        Dummy2:= copy(Line,9,5);   val(Dummy2,TT,Code);
        Dummy3:= copy(Line,27,6);  val(Dummy3,EqvP,Code);
        Dummy2:= copy(Line,42,5);  val(Dummy2,KIndex,Code);
        Dummy2:= copy(Line,60,5);  val(Dummy2,Sweat,Code);
   end;     { ReadInData }

{ ****************************************************************** }

   function Power(Base,Exponent: Real): Real;
   {
         Power of a number function.  Normally not defined in Pascal.
   }
   begin    { Power }
     Power:= exp(Exponent*ln(Base))
   end;     { Power }

{ ****************************************************************** }

   procedure ProcessRaob;
   {
      Processes the read-in data file and
      drives the main selection menu
   }
   const
     Kappa = 0.285856573;              { R/Cp }

   var
      U1, U2, V1, V2,           { u & v components used to calc shear}
      DH, AvgShear   : Real;    { Delta H & shear magnitude value    }
      Integrated,               { check for integrated hgts          }
      HgtinMeters,              { check for hght unit (m or mb       }
      FourKmShear,              { check for 4km shear calc only      }
      Combo          : Boolean; { check for KBD/Datafile input       }

   { =============================================================== }

      procedure ConvFromUV(U,V: Real;
                           var WindDir, WindSpd: Real);
      {
         Converts the u & v component of the wind to degrees
      }
```

```
const
  RTD = 57.29578122;

begin    { ConvFromUV }
   if (U < 0.0) and (V < 0.0) then       { Quadrant I   }
     begin
     end
   else
     if (U < 0.0) and (V > 0.0) then     { Quadrant II  }
       begin
       end
     else
       if (U > 0.0) and (V > 0.0) then   { Quadrant III }
         begin
         end
       else                              { Quadrant IV  }
         begin
         end
end;     { ConvFromUV }

procedure ConvtoUV(WindDir, WindSpd: Real;
                   var U,V: Real);
{
   Converts the input wind direction (in degrees) to the
   u & v components of the wind
}
const
  DTR = 1.74533E-02;      { Degree to radian conversion factor }

var
  Alpha: Real;                     { Quadrant angle to measure }

begin    { ConvtoU&V }
  if WindDir < 90 then                            { Quadrant I }
    begin
      Alpha:= 90 - WindDir;
      U:= -WindSpd*cos(DTR*Alpha);
      V:= -WindSpd*sin(DTR*Alpha)
    end
  else
    if ((WindDir >= 90)and(WindDir < 180)) then  { Quadrant II }
      begin
        Alpha:= WindDir - 90;
        U:= -WindSpd*cos(DTR*Alpha);
        V:=  WindSpd*sin(DTR*Alpha)
      end
    else
      if ((WindDir>=180)and(WindDir<270)) then  { Quadrant III }
        begin
          Alpha:= 270 - WindDir;
          U:= WindSpd*cos(DTR*Alpha);
```

```
             V:= WindSpd*sin(DTR*Alpha)
           end
         else                                          { Quadrant IV }
           begin
             Alpha:= WindDir - 270;
             U:=  WindSpd*cos(DTR*Alpha);
             V:= -WindSpd*sin(DTR*Alpha)
           end
   end;     { ConvtoU&V }

{ ================================================================ }

   procedure CalcAvgShear(DH,U1,V1,U2,V2: Real;
                          var AvgShear: Real);
   {
      Averages the wind shear in the layer (DH)
   }
   var
     DU,DV: Real;  { Differences in u & v components in the layer }

   begin    { CalcAvgShear }
     DU:= U2 - U1;
     DV:= V2 - V1;
     if Integrated then
       DH:= 1.0;
     AvgShear:= Sqrt(sqr(DU/DH) + sqr(DV/DH))
   end;     { CalcAvgShear }

{ ================================================================ }

   procedure FileShear(Combo: Boolean);
   {
       Computes the mean shear in the layer from either the input
       data file (not Combo option) or a combination of input data
       file and keyboard (Combo option)
   }
   type
      Intrp_Data = record                { Interpolated raob data }
                     WDir, WSpd,
                     Prs, Z     : Real
                   end;
      Intrp_Shr_Data = record            { Interpolated raob data }
                         WDir, WSpd,
                         Prs, Z     : Real
                       end;
      Intrp_Data_Array = array[1..200] of Intrp_Data;
      Intrp_Shr_Array  = array[1..50] of Intrp_Shr_Data;
      SearchIndex = 0..50;

   var
      DataIndex,              { Index in array for closest value    }
```

```
  Index1, Index2,
  Num_Intrp          : Integer;  { Number interpolated levels }
  Raw_Intp           : Intrp_Data_Array;
  Shr_Intp           : Intrp_Shr_Array;
  CheckWind,          { Key for missing data in winds (999.0) }
  Level1, Level2,
  SumAvgShear,
  WindDir1, WindSpd1,
  WindDir2, WindSpd2,
  WindSpd, WindDir,
  PLevel1, PLevel2,             { Pressure (mb) at levels 1/2 }
  ZLevel1, ZLevel2,             { Height (m) at levels 1 & 2  }
  StnElev            : Real;
  Winds,                        { Check for missing wind data }
  LevelError,                   { Check for proper level      }
  Continue           : Boolean; { Value for continuing calc   }

{ ----------------------------------------------------------- }

  function Search(Key: Real): SearchIndex;
  {
      Searches the data file for a match in the desired user
      input.  Does a binary search for the data in the raob
      structure until found.
  }
  type
    ExtRangeofSearchIndex = 0..51;

  var
    Check  : Real;
    Lower,                     { Lower bounds of search index }
    Upper,                     { Upper bounds of search index }
    Center : SearchIndex;      { Center of search index       }

  begin    { Search }
  {
     Initialize the search array bounds
    }
    Lower:= 1;
    Upper:= Num_Intrp;
    {
       Begin binary search of data
    }
    while Lower <= Upper do
      begin
        Center:= (Lower+Upper) div 2;
        {
           Search the height of the data in meters
        }
        if HgtinMeters then
          begin
      if Integrated then
       Check:= Shr_Intp[Center].Z
      else
       Check:= Raob[Center].Z_m;
      if Key = Check then
      {
        Z is found, get the hell outta Dodge
       }
       begin
        Search:= Center;
        exit
       end
      else
       if Key > Check then
       {
         Z is not found, so alter the
         search bounds accordingly
        }
        Upper:= Center - 1
       else
        Lower:= Center + 1
     end
    else
    {
      Search the height of the data in millibars
     }
     begin
      if Integrated then
       Check:= Shr_Intp[Center].Prs
      else
       Check:= Raob[Center].P_mb;
      if Key = Check then
       begin
        Search:= Center;
        exit
       end
      else
       if Key < Check then
        Lower:= Center + 1
       else
        Upper:= Center - 1
     end
   end;
  Search:= 0      { Data entry not found, so set index to 0 }
 end;    { Search }

{ ---------------------------------------------------------------- }

 procedure FindClosestValue(Key: Real;
                              var Indexer: Integer);
 {
   If an exact match of the desired levels is not found in
   the data file, then this procedure will find the closest
   level to the user-desired level
}
var
  Check       : Real;
  Count,                     { Raob array index counter     }
  Lower,Upper : SearchIndex; { Lower & upper array positions}
  Diff1,Diff2 : Real;        { Differences btwn two values  }
  Found       : Boolean;     { Successful search            }

begin    { FindClosestValue }
  Count:= 1;
  Found:= false;
  repeat
    if HgtinMeters then
      begin
        if Integrated then
          Check:= Shr_Intp[Count].Z
        else
          Check:= Raob[Count].Z_m;
        if Key > Check then
          Count:= Count + 1
        else
          begin
            Indexer:= Count;
            Found:= true
          end
      end
    else
      begin
        if Integrated then
          Check:= Shr_Intp[Count].Prs
        else
          Check:= Raob[Count].P_mb;
        if Key < Check then
          Count:= Count + 1
        else
          begin
            Indexer:= Count;
            Found:= true
          end
      end
  until (Found) or (Count = Num_Intrp);
  {
     Assign values to two adjacent values
     and compare for closest match
  }
  if Indexer <> 1 then    { Can't go below the first array }
    begin                 {   index so it must be found    }
      Lower:= Indexer-1;
    Upper:= Indexer;
    if HgtinMeters then
      begin
        if Integrated then
          begin
            Diff1:= abs(Shr_Intp[Lower].Z - Key);
            Diff2:= abs(Shr_Intp[Upper].Z - Key)
          end
        else
          begin
            Diff1:= abs(Raob[Lower].Z_m - Key);
            Diff2:= abs(Raob[Upper].Z_m - Key)
          end
      end
    else
      if Integrated then
        begin
          Diff1:= abs(Shr_Intp[Lower].Prs - Key);
          Diff2:= abs(Shr_intp[Upper].Prs - Key)
        end
      else
        begin
          Diff1:= abs(Raob[Lower].P_mb - Key);
          Diff2:= abs(Raob[Upper].P_mb - Key)
        end;
        {
           Assign index count to the closest fit in data
    }
    if Diff1 <= Diff2 then
      Indexer:= Lower
    else
      Indexer:= Upper
   end
 end;     { FindClosestValue }

{ ----------------------------------------------------------- }

 procedure Output_Results_File(Shr_Value: Real);
 {
    Outputs the shear results into an external data file
    chosen by the user.  If the data file already exists,
    it will append the output to the file.  If not, then
    it will create the file and output.
 }
 var
   Filename2: string[14];
   OutFile  : text;

 begin    { Output_Results_File }
   gotoxy(5,14); write(' ':74);
   gotoxy(5,14);
```

```
    write('What file name do you want for this file? ');
    readln(Filename2);
    {$I-}
    assign(Outfile,Filename2);
    append(OutFile);
    IOCode:= IOResult;
    if IOCode <> 0 then
      begin
        rewrite(OutFile);
        if Integrated then
          writeln(Outfile,' ':20,'Integrated shear results for
           ',Date,'---')
        else
          writeln(OutFile,' ':25,'Shear results for ',Date,
            '---');
        writeln(OutFile)
      end;
    {$I+}
    if Shr_Value <> 999.0 then
      begin
        write(OutFile,StnID,' at ',ZTime,'Z: ',Shr_Value,' m/s'
               ,' [ ');
        if Combo then
          write(OutFile,'  surface  ')
        else
          begin
            if Integrated then
              write(Outfile,Shr_Intp[Index1].Z:8:1,' m ')
            else
              write(OutFile,Raob[Index1].Z_m:8:1,' m ');
          end;
        if Integrated then
          write(OutFile,'to ',Shr_Intp[Index2].Z:8:1,' m ]')
        else
          write(OutFile,'to ',Raob[index2].Z_m:8:1,' m ]')
      end
    else
      write(OutFile,StnID,' at ',ZTime,'Z:  No shear data ',
             'available due to lack of wind profile data.');
    if FourKmShear then
      begin
        write(OutFile,' < 4 km shr>');
        writeln(OutFile)
      end
    else
      writeln(OutFile);
    close(OutFile);
  end;     { Output_Results_File }

{ -------------------------------------------------------------- }
```

```
procedure Meters_or_mbs;
{
    Asks the user whether the lowest 4 km shear is to be
    computed.  If not, then the user is prompted as to
    whether the requested levels in the layer to be evaluated
    is to be in meters or in millibars.  The boolean value is
    then sent to the calling procedure for subsequent compu-
    tations.
}
var
   YCount        : Integer; { Counter for blanking out menu }

begin    { Meters_or_Mbs }
{
    Blank out the menu box
  }
  TextColor(LightGreen);
  gotoxy(32,7);
   write(StnID,' - ',Date,'/',ZTime,'Z');
  TextColor(LightRed);
  for YCount:= 8 to 17 do
    begin
      gotoxy(4,YCount);
      write(' ':74)
    end;
  TextColor(LightRed);
  gotoxy(5,8);
  write('Do you want shear to be integrated every 200 m? (Y/N
   ): ');
  readln(Answer);
  if (Answer = 'Y') or (Answer = 'y') then
    Integrated:= true
  else
    Integrated:= false;
  gotoxy(5,8);
  if not Combo then
    begin
      write('Do you want to calculate the lowest 4km shear in
        the layer? (Y/N): ');
      readln(Answer)
    end
  else
    Answer:= 'N';

  if (Answer = 'Y') or (Answer = 'y') then
    begin
      Level1:= 1.0;
      FourKmShear:= true;
      HgtinMeters:= true
    end
  else
```

```
      begin
        gotoxy(5,8);
        for LCV:= 1 to 70 do
          write(' ');
        gotoxy(5,8);
        write('Do you want to input levels as meters (A) or mb
         (B): ');
        readln(Answer);
        if (Answer = 'A') or (Answer = 'a') then
          HgtinMeters:= true
        else
          HgtinMeters:= false
      end
  end;     { Meters_or_mbs }

{ ------------------------------------------------------------ }

  procedure Interp_Levels;
  {
     Linearly interpolates the data in the data field.
     First it interpolates every 5 mb; second it takes
     the values at every 200m (for dz=200) and places them
     into another data field for shear calculations if the
     shear to be calculated is integrated every 200 m.
  }
  var
    Z1, Z2,               { Adjacent hgt levels for close fit }
    Diff1, Diff2,          { Diff in hgt levels for close fit }
    Frac,                   { Linear interpolating fraction   }
    P_Intrp,                { Interpolated pressure level     }
    Mean_Depth_mb,          { depth of "low-level mean" layer }
    Intrp_Incr_mb : Real;   { Interpolation increment         }
    Index,                  { Index counter                   }
    Shr_Index,            { 200 m adjusted interpolated index }
    Data_Index,                { Data array index             }
    Intrp_Index   : Integer;      { Interpolation array index }
    Found_First   : Boolean;  { Check for finding 200 m level }

  begin    { Interp_Levels }
  {
      Determine the mean depth of the layer to 100 mb or
      at end of data (whichever comes first).  Set inter-
      polation increment to 5 mb.  Initialize indice pointers.
    }
    Mean_Depth_mb:= Raob[EndofData].P_mb;
    Intrp_Incr_mb:= 5.0;
    Intrp_Index:= 1;
    Data_Index:= 2;
    {
       Assign pressure in raob data structure and
       interpolate every 5 mb for the analyzed layer ...
}
P_Intrp:= Raob[1].P_mb;
while P_Intrp > Mean_Depth_mb do
 begin
  if P_Intrp < Raob[Data_Index].P_mb then
    Data_Index:= Data_Index + 1;
    {
         Calculate the interpolation fraction to use
         for future calculations
  }
  Frac:= (P_Intrp - Raob[Data_Index-1].P_mb) /
       (Raob[Data_Index].P_mb - Raob[Data_Index-1].P_mb);
  if Raob[Data_Index].Spd <> 999.0 then
   Raw_Intp[Intrp_Index].WSpd:= Raob[Data_Index-1].Spd +
    (Frac*(Raob[Data_Index].Spd - Raob[Data_Index-1].Spd))
  else
   Raw_Intp[Intrp_Index].WSpd:= 999.0;
  if Raob[Data_Index].Dir <> 999.0 then
   Raw_Intp[Intrp_Index].WDir:= Raob[Data_Index-1].Dir +
    (Frac*(Raob[Data_Index].Dir - Raob[Data_Index-1].Dir))
  else
   Raw_Intp[Intrp_Index].WDir:= 999.0;
  Raw_Intp[Intrp_Index].Z:= Raob[Data_Index-1].Z_m +
    (Frac*(Raob[Data_Index].Z_m - Raob[Data_Index-1].Z_m));
  Raw_Intp[Intrp_Index].Prs:= P_Intrp;
  Intrp_Index:= Intrp_Index + 1;
  P_Intrp:= P_Intrp - Intrp_Incr_mb
 end;
 {
     Now create a new array of interpolated values (~200 m)
     for use in the shear calculations.  If the first value
     is less than 200 m then assign that to the new array
     and adjust the index pointers accordingly.  If not,
     then set the index pointer to the first entry in the
     interpolated data set.
}
if Raw_Intp[1].Z <= 200.0 then
  begin
    Shr_Intp[1].Prs:= Raw_Intp[1].Prs;
    Shr_Intp[1].Z:= Raw_Intp[1].Z;
    Shr_Intp[1].WDir:= Raw_Intp[1].WDir;
    Shr_Intp[1].WSpd:= Raw_Intp[1].WSpd;
    Index:= 1;
    Shr_Index:= 2
  end
else
  begin
    Index:= 0;
    Shr_Index:= 1;
  end;
Num_Intrp:= Intrp_Index;
```

```
    Found_First:= false;
    {
       Find out where the lowest 200 m in the layer occurs in
       the interpolated data set.  If a close match occurs,
       then check which is closest to 200 m and assign it to
       the new data set.
    }
    repeat
      Index:= Index + 1;
      Z1:= Raw_Intp[Index].Z;   Z2:= Raw_Intp[Index+1].Z;
      if (Z1 >= 200.0) or (Z2 >= 200.0) then
        begin
          Diff1:= abs(200.0 - Z1);  Diff2:= abs(200.0 - Z2);
          if Diff1 < Diff2 then
            begin
              Shr_Intp[Shr_Index].Z:= Z1;
              Shr_Intp[Shr_Index].Prs:= Raw_Intp[Index].Prs;
              Shr_Intp[Shr_Index].WDir:= Raw_Intp[Index].WDir;
              Shr_Intp[Shr_Index].WSpd:= Raw_Intp[Index].WSpd
            end
          else
            begin
              Index:= Index + 1;
              Shr_Intp[Shr_Index].Z:= Z2;
              Shr_Intp[Shr_Index].Prs:= Raw_Intp[Index].Prs;
              Shr_Intp[Shr_Index].WDir:= Raw_Intp[Index].WDir;
              Shr_Intp[Shr_Index].WSpd:= Raw_Intp[Index].WSpd
            end;
          Found_First:= true
        end
    until Found_First;
    {
        Assign every 4th entry in the interpolated data set
        to the new data set, thus creating a new data set
        that is about 200 m apart in height.  Assign the
        number of data entries of the new array to Num_Intrp.
    }
    while Index+4 <= Num_Intrp do
      begin
        Shr_Index:= Shr_Index + 1;
        Index:= Index + 4;
        Shr_Intp[Shr_Index].Prs:= Raw_Intp[Index].Prs;
        Shr_Intp[Shr_Index].WDir:= Raw_Intp[Index].WDir;
        Shr_Intp[Shr_Index].WSpd:= Raw_Intp[Index].WSpd;
        Shr_Intp[Shr_Index].Z:= Raw_Intp[Index].Z;
      end;
    Num_Intrp:= Shr_Index
  end;     { Interp_Levels }

{ ------------------------------------------------------------ }
```

```
begin    { FileShear }
  if Raob[2].Spd <> 999.0 then
    Winds:= true
  else
    Winds:= false;
  if not Winds then
    begin
      gotoxy(5,12);
      write('Cannot perform shear calculations due to lack of ',
            'wind data in file.');
      gotoxy(5,14);
      write('Do you want to output this result to an external fi
       le? (Y/N): ');
      readln(Answer);
      if (Answer = 'Y') or (Answer = 'y') then
        Output_Results_File(999.0)
    end;
  if (not Combo) and Winds then
  {
        Data input via the input data file only
    }
    repeat
      FourKmShear:= false;
      Meters_or_Mbs;
      if Integrated then
        Interp_Levels;
      if not FourKmShear then
        repeat
        {
            Input levels via keyboard as
            either in millibars or meters
          }
          if HgtinMeters then
            begin
              gotoxy(5,8);
              write('What levels (m) for ',StnID,' do you want '
               ,ZTime,'Z shear computed for? ')
            end
          else
            begin
              gotoxy(5,8);
              write('What levels (mb) for ',StnID,' do you want
               ',ZTime,'Z shear computed for? ')
            end;
          readln(Level1,Level2);
          if HgtinMeters then
          {
              Data input levels to be in meters
            }
            begin
              if Level2 < Level1 then
       {
           Error in input
         }
         begin
           gotoxy(8,9);
           write(^G,'ERROR!! Bottom level greater than up
            per level');
           LevelError:= true
         end
       else
         begin
           gotoxy(8,9);
           write(' ':45);
           LevelError:= false
         end
     end
   else
   {
       Data input levels to be in millibars
     }
     begin
       if Level1 < Level2 then
       {
           Error in input
         }
         begin
           gotoxy(8,9);
           write(^G,'ERROR!! Bottom level less than upper
             level');
           LevelError:= true
         end
       else
         begin
           gotoxy(8,9);
           write(' ':45);
           LevelError:= false
         end
     end
  until not LevelError;
  {
     Output the appropriate answer and additional prompts
     for more info

     Find the array index for level 1.  If level1 is not
     in array, then find the closest value to that input
     level.
}
DataIndex:= Search(Level1);
if DataIndex <> 0 then
  Index1:= DataIndex
else
```

```
  FindClosestValue(Level1,Index1);
if FourKmShear then
  begin
    Level1:= Shr_Intp[Index1].Z;
    Level2:= Level1 + 4000.0
  end;
  {
       Find the array index for level 2.  If level2 is
       not in array, then find the closest value to
       that input level.
}
DataIndex:= Search(Level2);
if DataIndex <> 0 then
  Index2:= DataIndex
else
  FindClosestValue(Level2,Index2);
  {
     If the closest found values happen to be have the
     same index pointer, then increment index2 one
     position to avoid division by zero error.
}
if Index1 = Index2 then
  Index2:= Index1 + 1;
  {
     Convert the winds into the corresponding
     u & v coordinates

       Check for missing winds in the data and re-adjust
       the index array counter to where the data ends
       in the data array
}
if Integrated then
  CheckWind:= Shr_Intp[Index2].WDir
else
  CheckWind:= Raob[Index2].Dir;
if CheckWind = 999.0 then
  repeat
    Index2:= Index2 - 1;
    if Integrated then
      CheckWind:= Shr_Intp[Index2].WDir
    else
      CheckWind:= Raob[Index2].Dir
  until CheckWind <> 999.0;
if Integrated then
  begin
    SumAvgShear:= 0.0;
    LCV:= 1;
    repeat
      WindDir1:= Shr_Intp[LCV].WDir;
      WindSpd1:= Shr_Intp[LCV].WSpd;
      WindDir2:= Shr_Intp[LCV+1].WDir;
```

```
   WindSpd2:= Shr_Intp[LCV+1].WSpd;
   ConvtoUV(WindDir1,WindSpd1,U1,V1);
   ConvtoUV(WindDir2,WindSpd2,U2,V2);
   {
     Determine the height of the layer
   }
   CalcAvgShear(DH,U1,V1,U2,V2,AvgShear);
   SumAvgShear:= SumAvgShear + AvgShear;
   LCV:= LCV + 1
  until LCV = Index2;
  PLevel1:= Shr_Intp[Index1].Prs;
  ZLevel1:= Shr_Intp[Index1].Z;
  PLevel2:= Shr_Intp[LCV].Prs;
  ZLevel2:= Shr_Intp[LCV].Z;
  DH:= ZLevel2 - ZLevel1;
  AvgShear:= SumAvgShear/DH
 end
else
 begin
  ConvtoUV(Raob[Index1].Dir,Raob[Index1].Spd,U1,V1);
  ConvtoUV(Raob[Index2].Dir,Raob[Index2].Spd,U2,V2);
  DH:= Raob[Index2].Z_m - Raob[Index1].Z_m;
  CalcAvgShear(DH,U1,V1,U2,V2,AvgShear);
  PLevel1:= Raob[Index1].P_mb;
  ZLevel1:= Raob[Index1].Z_m;
  PLevel2:= Raob[Index2].P_mb;
  ZLevel2:= Raob[Index2].Z_m
 end;
gotoxy(5,10);
write('The avg shear magnitude from ',PLevel1:4:0,' mb [ '
 ,ZLevel1:7:1,' m] ');
gotoxy(9,11);
write('to ',PLevel2:4:0,' mb [ ',ZLevel2:7:1,' m] is:');
TextColor(LightGreen);
gotoxy(5,12);
write('    ',AvgShear,' per sec.');
TextColor(LightRed);
gotoxy(5,14);
write('Hit any key to continue....');
repeat until Keypressed;
 Answer:= Readkey;
gotoxy(5,14);
write('Do you want to output these results to an external
 file? (Y/N): ');
readln(Answer);
if (Answer = 'Y') or (Answer = 'y') then
 Output_Results_File(AvgShear);
gotoxy(5,14);
write('Want to do another shear calculation for ',StnID,'?
 (Y/N):                ');
gotoxy(60,14);
```

```
    readln(Answer);
    if (Answer = 'Y') or (Answer = 'y') then
      Continue:= true
    else
      Continue:= false
  until not Continue
else
  if Winds then
  {
       Input levels via data file and keyboard
   }
   repeat
     repeat
       FourKmShear:= false;
       Meters_or_Mbs;
       if Integrated then
         Interp_Levels;
       gotoxy(5,8);
       write('What are the surface winds (direction & speed)?
              ');
       gotoxy(53,8);
       readln(WindDir,WindSpd);
       if (WindDir < 0 ) or (WindDir > 360.0) then
         begin
           gotoxy(5,9);
           write(^G,'ERROR!! Invalid wind direction ---> ',Win
            dDIr:6:1);
           LevelError:= true
         end
       else
         LevelError:= false
     until Not LevelError;
     gotoxy(5,9);
     write(' ':70);
     gotoxy(5,9);
     write('What is the station height (in meters) of ',StnID,
      '? ');
     readln(StnElev);
     gotoxy(5,10);
     if HgtinMeters then
      write('Up to what level (in meters) do you want to compu
       te shear? ')
     else
      write('Up to what level (in mb) do you want to compute s
       hear? ');
     readln(Level2);
     {
         Find the array index for level 2.  If level2 is not
         in array, then find the closest value to that input
         level.
     }
```

```
Level1:= Shr_Intp[2].Z;
if HgtinMeters then
  Level2:= StnElev + Level2;
DataIndex:= Search(Level2);
if DataIndex <> 0 then
  Index2:= DataIndex
else
  FindClosestValue(Level2,Index2):
  {
     Convert the winds into the corresponding u & v
     coordinates
}
if Integrated then
  CheckWind:= Shr_Intp[Index2].WDir
else
  CheckWind:= Raob[Index2].Dir;
if CheckWind = 999.0 then
  repeat
    Index2:= Index2 - 1;
    if Integrated then
      CheckWind:= Shr_Intp[Index2].WDir
    else
      CheckWind:= Raob[Index2].Dir
  until CheckWind <> 999.0;
if Integrated then
  begin
    ConvtoUV(WindDir,WindSpd,U1,V1);
    ConvtoUV(Shr_Intp[2].WDir,Shr_Intp[2].WSpd,U2,V2);
    {
        Determine the height of the bottom layer (200 m)
    }
    DH:= Level1 - StnElev;
    CalcAvgShear(DH,U1,V1,U2,V2,SumAvgShear);
    LCV:= 2;
    repeat
      WindDir1:= Shr_Intp[LCV].WDir;
      WindSpd1:= Shr_Intp[LCV].WSpd;
      WindDir2:= Shr_Intp[LCV+1].WDir;
      WindSpd2:= Shr_Intp[LCV+1].WSpd;
      ConvtoUV(WindDir1,WindSpd1,U1,V1);
      ConvtoUV(WindDir2,WindSpd2,U2,V2);
      {
          Determine the height of the layer for
          integration
      }
      CalcAvgShear(DH,U1,V1,U2,V2,AvgShear);
      SumAvgShear:= SumAvgShear + AvgShear;
      LCV:= LCV + 1
    until (LCV = Index2) or not Winds;
    PLevel2:= Shr_Intp[LCV].Prs;
    ZLevel2:= Shr_Intp[LCV].Z;
```

```
              DH:= ZLevel2 - StnElev;
              AvgShear:= SumAvgShear/DH
            end
          else
            begin
              ConvtoUV(WindDir,WindSpd,U1,V1);
              ConvtoUV(Raob[Index2].Dir,Raob[Index2].Spd,U2,V2);
              DH:= Raob[Index2].Z_m - StnElev;
              CalcAvgShear(DH,U1,V1,U2,V2,AvgShear);
              PLevel2:= Raob[Index2].P_mb;
              ZLevel2:= Raob[Index2].Z_m
            end;
          gotoxy(5,11);
          write('The avg shear magnitude from sfc of ',StnID,
                ' to ',PLevel2:4:0,' mb [ ',ZLevel2:7:1,' m] is:');
          TextColor(LightGreen);
          gotoxy(5,12);
          write('    ',AvgShear,' per sec.');
          TextColor(LightRed);
          gotoxy(5,14);
          write('Hit any key to continue....');
          repeat until Keypressed;
           Answer:= Readkey;
          gotoxy(5,14);
          write('Do you want to output these results to an external
            file? (Y/N): ');
          readln(Answer);
          if (Answer = 'Y') or (Answer = 'y') then
            Output_Results_File(AvgShear);
          gotoxy(5,14);
          write('Want to do another shear calculation for ',StnID,'
           ? (Y/N):           ');
          gotoxy(60,14);
          readln(Answer);
          if (Answer = 'Y') or (Answer = 'y') then
            Continue:= true
          else
            Continue:= false
        until not Continue
   end;     { FileShear }

{ ============================================================== }

   procedure InputDataComputeShear;
   {
       Computes wind shear by keyboard inputs only.
   }
   var
     YCount   : Integer;
     WindSpd,
     WindDir,
  Z1, Z2   : Real;
  Again    : Boolean;       { Check for a rerun of shear comps }

begin    { InputDataComputeShear }
  FourKmShear:= false;
  StnId:= 'input';
  repeat
    repeat
    {
         Blank out the menu box
      }
      for YCount:= 8 to 17 do
        begin
          gotoxy(4,YCount);
          write(' ':74)
        end;
      TextColor(LightRed);
      gotoxy(6,10);
      write('Input the wind (deg & spd) and hgt (meters) for lev
       el 1: ');
      readln(WindDir,WindSpd,Z1);
      if (WindDir < 0.0) or (WindDir > 360) then
        begin
          GoodInput:= false;
          gotoxy(8,11);
          write(^G,'ERROR!! Wind direction is undefined ---> ',W
           indDir:6:1)
        end
      else
        GoodInput:= true
    until GoodInput;
    ConvtoUV(WindDir,WindSpd,U1,V1);
    repeat
      gotoxy(6,11);
      write('Input the wind (deg & spd) and hgt (meters) for lev
       el 2: ');
      readln(WindDir,WindSpd,Z2);
      if (WindDir < 0.0) or (WindDir > 360.0) then
        begin
          GoodInput:= false;
          gotoxy(8,12);
          write(^G,'ERROR!! Wind direction is undefined ---> ',W
           indDir:6:1)
        end
      else
        GoodInput:= true
    until GoodInput;
    ConvtoUV(WindDir,WindSpd,U2,V2);
    {
       Determine the depth of the layer
    }
```

```
       DH:= Z2 - Z1;
       CalcAvgShear(DH,U1,V1,U2,U2,AvgShear);

       TextColor(Yellow);
       gotoxy(5,12);
       {
          Output the answer
       }
       write('The average shear magnitude is: ',AvgShear,
             ' per sec.');
       TextColor(LightRed);
       gotoxy(5,14);
       write('Hit any key to continue....');
       repeat until Keypressed;
       Answer:= Readkey;
       gotoxy(5,14);
       write('Do you want to input another pair of winds? (Y/N) ');
       readln(Answer);
       if (Answer = 'Y') or (Answer ='y') then
         Again:= true
       else
         Again:= false
     until not Again
   end;     { InputDataComputeShear }

{ =============================================================== }


   procedure PlotRaob;
   {
       Plots on the EGA CRT a Skew-T with Pressure, Temperature,
       & Adiabats a sounding based upon the input McIdas data file.
       Uses previous Raob data structure.
   }
   var
     GrDriver,                      { Graphics device driver       }
     GraphMode,                     { Graphics mode value - EGA    }
     ErrorCode,                     { Reports any graphics errors  }
     TAdiabat,                      { Pot Temp for plot adiabats   }
     Colon,                         { Colon position in file name  }
     Testy,                         { Mod test for dashed lines    }
     Xabs, Yabs,                    { X & Y coordinates for plot   }
     X1, X2, Y1, Y2,                {   "        "       "    "    }
     X, Y, X1P, X2P   : Integer;    {   "        "       "    "    }
     Temp, T,
     PS, DP, DP2      : Real;       { File temp & press values     }
     PLabel           : String[6];  { Label for pressure axis      }
     Dummy            : String[14]; { Dummy file name for EGA      }
     DataFileName     : String[8];  { Output file name for EGA     }

{ --------------------------------------------------------------- }
```

```
     function Ordinate(T,P: Real): Integer;
     {
         Returns an ordinate value along the x-axis for a given
         pair of input values
     }
     begin    { Ordinate }
       Ordinate:= round(40+8*(T+35+32.74*(ln(1000)-ln(P))))
     end;     { Ordinate }

{ --------------------------------------------------------------- }

     function Absisca(P: Real): Integer;
     {
         Returns an absisca value along the y-axis for an
         input value to plot along the vertical
     }
     begin    { Absisca }
       Absisca:= round(310 - 250.8 * (ln(1000) - ln(P)))
     end;     { Absisca }

{ --------------------------------------------------------------- }

     function Press(Y: Real): Real;
     {
         Returns a value in pixel units for plotting when
         given a value in pressure units (mb)
     }
     begin    { Press }
       Press:= exp((Y - 310)/250.8 + ln(1000))
     end;     { Press }


   begin    { PlotRaob }
     ClrScr;
     {
         Initialize graphics card and check for incompatability
     }
     GrDriver:= Detect;
     InitGraph(GrDriver,GraphMode,'');
     ErrorCode:= GraphResult;
     if ErrorCode <> grOK then
       begin
         writeln(^G,'GRAPHICS ERROR: ',GraphErrorMsg(ErrorCode));
         writeln('Program aborted...');
         repeat until Keypressed; Answer:= ReadKey;
         Halt(1)
       end;
     {
        Draw the box around the sounding and label the legend
     }
```

```
SetColor(LightCyan);
Rectangle(37,5,603,313);    Rectangle(38,6,602,312);
{
   If the data file is located on another disk drive, then
   take out the drive specification and colon and rename
   the data file as the file name itself without the specifi-
   cation
}
Dummy:= Filename;
Colon:= pos(':',Dummy);
if Colon = 0 then       { No specification thus Ok file name }
  DataFileName:= Filename
else                        { Delete the drive specification }
  begin
    Delete(Dummy,1,Colon);
    DataFileName:= Dummy
  end;
SetColor(Yellow);
Rectangle(150,339,495,325);
OutTextXY(155,329,'Upper Air Sounding Plot For');
OutTextXY(380,329,StnID);    OutTextXY(430,329,'at');
OutTextXY(450,329,ZTime);    OutTextXY(485,329,'Z');
Line(39,339,80,339);         OutTextXY(40,329,Date);
Line(548,339,608,339);       OutTextXY(549,329,DataFileName);
{
   Label and plot pressure axes
}
SetColor(LightCyan);
OutTextXY(1,308,'1000');
X:= 40;   Y:= 600;
PS:= 900;               { 900 mb }
repeat
  Str(PS:4:1,PLabel);
  Delete(PLabel,4,2);
  Yabs:= Absisca(PS);
  OutTextXY(1,Yabs-2,PLabel);
  Line(X-1,Yabs,Y+1,Yabs);
  PS:= PS - 100         { 800 - 300 mb }
until PS = 300;
OutTextXY(1,4,'300');
{
   Plot temperature axis
}
SetColor(LightRed);
Line(560,310,600,275);     {  30 deg isotherm }
Line(480,310,600,205);     {  20 deg isotherm }
Line(400,310,600,135);     {  10 deg isotherm }
Line(320,310,600,65);      {   0 deg isotherm }
Line(240,310,580,8);       { -10 deg isotherm }
Line(160,310,500,8);       { -20 deg isotherm }
Line(80,310,420,8);        { -30 deg isotherm }
Line(40,270,335,8);        { -40 deg isotherm }
Line(40,195,250,8);        { -50 deg isotherm }
Line(40,117,165,8);        { -60 deg isotherm }
{
   Label temperature axes
}
OutTextXY(550,316,'30');   OutTextXY(470,316,'20');
OutTextXY(390,316,'10');   OutTextXY(315,316,'0');
OutTextXY(227,316,'-10');  OutTextXY(147,316,'-20');
OutTextXY(67,316,'-30');   OutTextXY(10,270,'-40');
OutTextXY(10,195,'-50');   OutTextXY(10,115,'-60');
{
   Plot Dry Adiabats
}
TAdiabat:= 30;
repeat
  T:= TAdiabat;
  PS:= 1000;
  Xabs:= ordinate(T,PS);
  Yabs:= absisca(PS);
  repeat
    if Xabs <= 600 then
      putpixel(Xabs,Yabs,Yellow);
    Yabs:= Yabs - 1;
    {
       Draw every four pixels for dashed effect
    }
    TestY:= Yabs mod 4;
    if TestY = 0 then
      Yabs:= Yabs - 4;
    PS:= Press(Yabs);
    {
       Compute dry adiabat
    }
    T:= (Power((PS/1000),Kappa)*(TAdiabat+273.15))-273.15;
    Xabs:= ordinate(T,PS)
  until ((Xabs < 40) or (Yabs < 8));  { Top and left of screen
    }
  if Yabs <= 8 then
    T:= TAdiabat + 273;
  TAdiabat:= TAdiabat - 10
until TAdiabat < -35;
{
    Plot Raob data from file

    Display pressure level of first data point
}
SetColor(LightGreen);
Str(Raob[1].P_mb:6:1,PLabel);
Delete(PLabel,4,2);
Yabs:= Absisca(Raob[1].P_mb);
```

```
     OutTextXY(604,Yabs-2,PLabel);
     LCV:= 1;
     repeat
       DP:= Raob[LCV].TD_C;       { Dewpoint            }
       PS:= Raob[LCV].P_mb;       { Pressure level      }
       Temp:= Raob[LCV].T_C;      { Parcel temperature }
       LCV:= LCV + 1;
       X1:= ordinate(Temp,PS);
       Y1:= absisca(PS);

       X2:= ordinate(Raob[LCV].T_C,Raob[LCV].P_mb);
       Y2:= absisca(Raob[LCV].P_mb);

       DP2:= Raob[LCV].TD_C;
       X1P:= ordinate(DP,PS);
       X2P:= ordinate(DP2,Raob[LCV].P_mb);

       if Y2 >= 8 then  { point is at or below top of border -- Ok}
         begin
           SetLineStyle(SolidLn,SolidFill,NormWidth);
           Line(X1,Y1,X2,Y2);
           SetLineStyle(DashedLn,SolidFill,NormWidth);
           if DP2 <> 999.0 then    { Don't plot missing values }
             Line(X1P,Y1,X2P,Y2)
         end;
       PS:= Raob[LCV].P_mb;
       Temp:= Raob[LCV].T_C;
       DP:=DP2
     until ((LCV = EndofData) or (Y2 <= 8));
     repeat until KeyPressed;
     Answer:= ReadKey;
     RestoreCrtMode
   end;     { PlotRaob }

{ ================================================================ }

procedure ConvertDataFile;
{
    Converts McIDAS Data file into a format needed for PBE
    program (on MEAVAX). Procedure takes pressure, temperature,
    and dew point and creates a different file chosen by the
    user.
}
const
  Keyword100 = ' 100.0';  {       End of data up to 100 mb        }
  Keyword    = 'PARCEL';  { Denotes start of pre-determined paras }

type
  FileData = record                      { Raob data structure       }
                P_mb      : Integer;
                T_C,TD_C  : Real
             end;
  DummyStr = string[6];

var
  PBERaob     : FileData;
  Outfilename : string[50];
  Line        : string[80];  { Single read-in line from data file }
  Dummy1      : string[4];   { Dummy variables for individual     }
  Dummy2      : string[5];   {   data entries in line             }
  Check,                     { Checks for word match in data      }
  Dummy3      : DummyStr;    { Dummy variables for individual     }
  Dummy4      : string[7];   {   data entries in line             }
  DumVar      : LongInt;     { Dummy variable for Station ID      }
  Code        : Integer;     { Error code for string functions    }
  Result      : Real;        { Corrected data entry for program   }
  ConvertNoMore,             { Check to end conversion process    }
  Found,                     { Boolean for end of data field      }
  Missing     : Boolean;     { Boolean for missing leading zeroes }
  Outfile,                   { Output file converted data is put  }
  Datafile    : text;        { Input file name                    }

   begin    { ConvertDataFile }
      TextColor(LightRed);
      {
         Blank out the menu box contents
      }
      repeat
        for LCV:= 8 to 17 do
          begin
            gotoxy(4,LCV);
            write(' ':74)
          end;
          {
             Test whether the file exists and
             display message if file doesn't
        }
        {$I-}
        repeat
          gotoxy(5,8);
          write('What filename does the data belong?: ');
          readln(Filename);
          assign(Datafile,Filename);
          reset(Datafile);
          IOCode:= IOResult;
          if IOCode <> 0 then
           begin
            gotoxy(7,9);
            writeln(^G,'ERROR! File ',Filename,' does not exist.',
                    ' Please re-enter...')
           end
        until IOCode = 0;
```

```
{$I+}
gotoxy(5,9); write(' ':70);
{
  Read in first line of data
   & determine where the data ends
}
Found:= false;
readln(DataFile,Line);
gotoxy(5,8);
writeln('Reading in ',filename,
        ' data...                       ');
{
   Extract remaining pertinent information

   Read in data up to 100 mb or when data ends
}
gotoxy(5,10);
write('What file name do you want for converted file?: ');
readln(Outfilename);
assign(outfile,Outfilename);
rewrite(outfile);

Found:= false;
repeat
  readln(Datafile,Line);
  Check:= copy(Line,2,6);
  if Check <> Keyword then
    begin
      Dummy3:= Check;
      val(Dummy3,Result,Code);
      PBERaob.P_mb:= round(Result);
      Dummy3:= copy(Line,11,5);
      InsertZero(Dummy3,Result);
      PBERaob.T_C:= Result;
      Dummy3:= copy(Line,19,5);
      CheckMissingData(Dummy3,Missing);
      if not Missing then
        begin
          InsertZero(Dummy3,Result);
          PBERaob.TD_C:= Result
        end
      else
        PBERaob.TD_C:= -99.0;
      writeln(outfile,PBERaob.P_mb,' ',PBERaob.T_C:5:1,' ',
               PBERaob.TD_C:5:1)
    end;
  if (PBERaob.P_mb <= 100) or (Check = Keyword) then
    Found:= true
until Found;
close(outfile);
gotoxy(5,12);
```

```
          write('Do you want to convert another file? (Y/N): ');
        readln(Answer);
        if (Answer = 'Y') or (Answer = 'y') then
          ConvertNoMore:= false
        else
          ConvertNoMore:= true
      until ConvertNoMore
   end;     { ConvertDataFile }

  { ============================================================ }

begin   { ProcessRaob }
   Choice:= 0;
   BoxOutline;
   gotoxy(2,7);
   write(#$C7);
   for LCV:= 1 to 76 do
     write(#$C4);
   write(#$B6);

   TextColor(LightGreen);
   gotoxy(11,6);
   write('WELCOME TO THE EXCITING WORLD OF UPPER AIR DATA PROCESSIN
    G!');
   TextColor(LightMagenta);
   gotoxy(14,8);
   write('What do you want to do? (Pick a topic) ___');
   TextColor(LightRed);
   gotoxy(8,10);
   write('(1) - Calculate wind shear from McIDAS raob file');
   gotoxy(8,11);
   write('(2) - Calculate wind shear by combination of McIDAS');
   gotoxy(8,12);
   write('         raob datafile & keyboard input');
   gotoxy(8,13);
   write('(3) - Calculate shear by keyboard input of winds & height
    s (m)');
   gotoxy(8,14);
   write('(4) - Plot sounding from McIDAS raob file');
   gotoxy(8,15);
   write('(5) - Convert McIDAS data files into PBE format');
   gotoxy(8,16);
   write('(6) - Quit Program');

   TestifFirstRun:= TestifFirstRun + 1;

   gotoxy(54,8);
   readln(Choice);
   case Choice of
            1 : begin
                  BoxOutline;
      TextColor(Yellow);
      gotoxy(30,6);
      write('WIND SHEAR CALCULATIONS');
      Initialize;
      ReadInData;
      AlreadyReadInData:= true;
      Combo:= false;
      FileShear(Combo)
    end;
2 : begin
      BoxOutline;
      TextColor(Yellow);
      gotoxy(30,6);
      write('WIND SHEAR CALCULATIONS');
      Initialize;
      ReadInData;
      AlreadyReadInData:= true;
      Combo:= true;
      FileShear(Combo)
    end;
3 : begin
      BoxOutline;
      TextColor(Yellow);
      gotoxy(25,6);
      write('KEYBOARD WIND SHEAR CALCULATIONS');
      InputDataComputeShear
    end;
4 : begin  { Plot sounding }
      if TestifFirstRun = 1 then
        AlreadyReadInData:= false;
      BoxOutline;
      TextColor(Yellow);
      gotoxy(22,6);
      write('PLOT SOUNDING FROM McIDAS DATA FILE');
      if AlreadyReadInData then
        begin
          TextColor(LightRed);
          gotoxy(5,10);
          write('Do you want to use the existing data',
                ' file? (Y/N): ');
          readln(Answer);
          if (Answer = 'N') or (Answer = 'n') then
            begin
              Initialize;
              ReadInData
            end
        end
      else
        begin
          Initialize;
          ReadInData
```

```
                       end;
                     PlotRaob;
                     BoxOutline;
                     TextColor(Yellow);
                     gotoxy(22,6);
                     write('PLOT SOUNDING FROM McIDAS DATA FILE');
                     TextColor(LightRed);
                     AlreadyReadInData:= false
                   end;
               5 : begin
                     BoxOutline;
                     TextColor(Yellow);
                     gotoxy(21,6);
                     write('CONVERT McIDAS DATA FILE TO PBE FORMAT');
                     TextColor(LightRed);
                     ConvertDataFile
                   end;
               6 : begin
                     ClrScr;
                     halt
                   end
      end    { case }
   end;     { ProcessRaob }

{ ****************************************************************** }

   procedure WrapUp(var Over: Boolean);

   begin    { WrapUp}
     gotoxy(5,15);
     write('Do you want to process other parameters in the program?',
            ' (Y/N) ');
     readln(Answer);
     if (Answer = 'Y') or (Answer = 'y') then
       Over:= false                       { Continue running program  }
     else
       Over:= true                     { or halt execution of program }
   end;    { WrapUp}
{
   -------------------------- MAIN PROGRAM --------------------------
}
begin
  ProgramInfo;
  TestifFirstRun:= 0;       { Initially a first run of the  program }
  repeat
    ProcessRaob;
    WrapUp(Over)
  until Over;
  ClrScr
end.
```

Appendix 6.4

PBE Program Listing

```
#include <stdio.h>
#include <math.h>
#define  YES           1
#define  NO            0
#define  intrp_incr_Pa  500.0  /* Interpolation Increment, Pa          */

/* This program computes CAPE in several forms; it follows similar  */
/* procedures to a routine written by E. Rasmussen at the University*/
/* of Illinois in 1983 to compute Potential Buoyant Energy (PBE).   */
/* There are several possible methods of computing PBE or CAPE:     */
/*   lifting a parcel having potential temperature and mixing ratio */
/*      properties similar to air near the ground as determined     */
/*      by a sounding, to the LCL and then to the LFC above which   */
/*      CAPE is summed.                                             */
/*   assuming low-level conditions are those of a mixed layer, with */
/*      the top of the mixed layer being at the CCL/LFC.  The       */
/*      mixed layer is represented on a sounding by the inter-      */
/*      section of the mean low-level mixing ratio with the temp    */
/*      sounding; mixed-layer temperature then being given by       */
/*      constant potential temperature from this level.  This is    */
/*      closely approximated at the time of maximum temperature;    */
/*      this is much more representative than the morning lifted-   */
/*      parcel technique, since convection often peaks at that time */
/*   using low-level conditions represented by surface temperature  */
/*      and dewpoint at the time of the onset of convection, and    */
/*      assuming the mixing ratio (computed from the dewpoint) is   */
/*      represetative of the parcels feeding the convection         */
/*                                                                  */
/* This routine will compute CAPE (PBE) through the lifting method. */
/*                                                                  */
/* Techniques for computation were designed based on the equivalent */
/* graphical techniques on thermodynamic diagrams.                  */
/*                                                                  */
/*    Erik Rasmussen                                                */
/*    Colorado State University  1/89                               */
/*                                                                  */
/* Minor modifications were done to Erik's original program.  Only  */
/*    to input/output and one conditional was added to the interpo- */
/*    lation routine concerning missing Td values (-999).  Other-   */
/*    wise the program is intact as written by Erik.                */
/* Attempts to implement this program on a PC were uneventful.      */
/*    Currently program resides on MEAVAX, and will run on any      */
/*    VMS/VAX based machine.                                        */
/*                                                                  */
/*    Dale R. Perry                                                 */
/*    North Carolina State University  3/89                         */
```

```
FILE   *fopen(),
       *outfile;                 /* Pointer to output file             */

int    num_raw_lvls,             /* Number of input data levels        */
       raw_data_index,           /* Counter in raw data arrays         */
       len,                      /* Output filename length             */
       intrp_index,              /* Counter in interpolated arrays     */
       CCL_found,                /* Flag for CCL                       */
       LCL_found,                /* Flag for LCL                       */
       i,                        /* Generic counter                    */
       p_raw_mb[100];            /* Input pressures, mb                */

double t_raw_C[100],             /* Input temperatures                 */
       td_raw_C[100],            /* Input dew points                   */
       w,                        /* Mixing ratio, dimensionless        */
       theta,                    /* Potential temperature, K           */
       frac,                     /* Linear interpolating fraction      */
       p_intrp,                  /* Interpolated pressure              */
       sum_w,                    /* Summed mixing ratio                */
       sum_theta,                /* Summed potential temperature       */
       mean_w,                   /* Mean mixing ratio in 100 mb layer  */
       mean_theta,               /* Mean theta in 100 mb layer         */
       t_on_mn_theta,            /* T of pot temp on mean_theta adbt   */
       theta_CCL,                /* Potential temperature at CCL       */
       theta_e_CCL,              /* Equivalent potential temperature   */
                                 /*  of a parcel originally at top of  */
                                 /*  the mixed layer.                  */
       theta_e_LCL,              /* Equivalent potential temperature   */
                                    /*  of a parcel brought to saturation*/
                                 /*  by lifting.                       */
       parcel_t_K,               /* Temperature of a parcel ascending  */
                                 /*  at a constant theta_e             */
       conv_temp_K,              /* Convective temperature, K          */
       sum_neg_area,             /* Negative area on thermo diagram    */
       pos_area,                 /* Positive area in a layer           */
       cape,                     /* Positive area, PBE, CAPE           */
       g = 9.807,                /* Gravitation constant               */
       mean_depth_Pa,            /* Depth of "low-level mean" layer    */
       depth_low_level,          /* Depth of low-level 400 mb layer    */
       p_CCL_Pa,                 /* Pressure at CCL                    */
       p_LCL_Pa,                 /* Pressure at LCL                    */
       p_int_Pa[200],            /* Interpolated pressure, Pa          */
       t_int_K[200],             /* Interpolated temperature, K        */
       td_int_K,                 /* Interpolated dew point temp, K     */
       w_int[200],               /* Interpolated mixing ratio          */
       delta_z_m[200],           /* Hydrostatic delta-z, meters        */
       z_m,                      /* Height, m                          */
       p_raw_Pa[100],            /* Input pressure in Pa               */
       t_raw_K[100],             /* Input temperature in K             */
       td_raw_K[100];            /* Input dew point temp in K          */
```

```
char   data_file_nm[31],       /* Data file name for input         */
       *file_out,              /* Pointer to output file           */
       *file_ext;              /* Pointer to output file extension */

double K_to_C(),               /* Converts degree K to C           */
       C_to_K(),               /* Converts degree C to K           */
       mb_to_Pa(),             /* Converts pressure mb to Pa       */
       poisson_t(),            /* Finds T given P, Theta           */
       poisson_p(),            /* Finds P given T, Theta           */
       poisson_theta(),        /* Finds Theta given T, P           */
       mix_rat(),              /* Finds mixing ratio               */
       vap_press(),            /* Finds vapor pressure, Pa         */
       theta_e(),              /* Computes equivalent pot temp     */
       t_of_theta_e();         /* Computes T given theta_e         */


input_ua(filename)        /* Routine to get raw UA data from a file */
  char *filename;
  {
  FILE *fopen(),
       *infile;                /* Pointer to input data file       */

  int  num_raw_lvls,           /* Number of read in data levels    */
       f,                      /* Return from an fscanf()          */
       i;                      /* Level counter                    */
  extern int p_raw_mb[100];    /* Input pressure data, mb          */
  extern double
       t_raw_C[100],           /* Input temperature, C             */
       td_raw_C[100];          /* Input dew point, C               */

  printf("Data: %s\n",filename);
  if((infile = fopen(filename,"r")) == NULL)
     {
     fprintf(stderr,"Unable to open %s for input\n",filename);
     return(-1);
     }
  i=0;
  while(fscanf(infile,"%d %f %f\n",&p_raw_mb[i],&t_raw_C[i],&td_raw_C[
   i])
        != EOF)i++;

  num_raw_lvls = i;

  return(num_raw_lvls);
  }


  double C_to_K(double t_C)
    {
    return(t_C + 273.16);      /* C ---> K, return.....            */
  }


double K_to_C(double t_K)
  {
  return(t_K - 273.16);      /* K ---> C, return.....            */
  }


double vap_press(double t_K)
  {
  double e_mb;               /* Vapor pressure, mb; converted to */
                             /*  just prior to return()          */

                             /* Uses Magnus' formula (see        */
                             /* Iribarne & Godson, p. 63)        */
                             /* Good to > significant figures    */

                             /* NOTE: pow(10,...) accomplishes   */
                             /* antilog; also that the last term */
                             /* is incorrect in Tribarne & Godson */

  e_mb = pow(10.0,((-2937.4/t_K) - (4.9283*log10(t_K)) + 23.5471));
  return(e_mb*100.0);        /* mb ---> Pa, return.....          */
  }


double mix_rat(double t_K, double p_Pa)
  {
  double w,                  /* Mixing ratio, unitless; used     */
                             /*  for both actual and saturated   */
         e_Pa;               /* Vapor pressure, Pa               */
  double vap_press();
                             /* Uses w = epsilon*e/(p-e)         */
                             /* Call on vap_press() for e        */
  e_Pa = vap_press(t_K);
  w = (0.622 * e_Pa)/(p_Pa - e_Pa);
  return(w);
  }


double mb_to_Pa(int p_mb)
  {
  return(p_mb * 100.0);      /* mb ---> Pa, return.....          */
  }


double poisson_p(double theta, double t_K)
  {
  double p_Pa,               /* Pressure, Pa                     */
         invrs_kappa;        /* Cp/R = 1/Kappa                   */
```

```
                             /* Use Poisson's equation to find    */
                             /*  pressure given pot temp & T      */

  invrs_kappa = 3.496503;
  p_Pa = 100000.0 * exp(invrs_kappa * log(t_K/theta));
  return(p_Pa);
  }


double poisson_t(double theta, double p_Pa)
  {
  double t_K;                /* Temperature, K                    */
                             /* Uses Poisson's equation to find   */
                             /*  temperature given a pot temp & P */

  t_K = theta * pow((p_Pa/100000.),0.286);
  return(t_K);
  }


double poisson_theta(double t_K, double p_Pa)
  {
  double theta;              /* Potential temperature, K          */
                             /* Uses Poisson's equation to find   */
                             /*  pot temp given a temperature & P */

  theta = (t_K * pow((100000./p_Pa),0.286));
  return(theta);
  }


double theta_e(double t_K, double p_Pa)
  {
  double w,                  /* Mixing ratio, unitless; used for  */
                             /*  both actual and saturated        */
         theta,              /* Potential temperature, K          */
         theta_e;            /* Equivalent potential temperature  */
  double mix_rat(),
         poisson_theta();
                             /* Equiv pot temp = pot temp *       */
                             /*  exp(L*sat mix rat/(Cp*T))        */

  w = mix_rat(t_K,p_Pa);
  theta = poisson_theta(t_K,p_Pa);
  theta_e = theta * exp((2.501E6 * w)/(1005.0 * t_K));
  return(theta_e);
  }


double t_of_theta_e(double t_lwr_bnd_K, double p_Pa,
                    double theta_e_knwn)
```

```
    {
                        /* t_lwr_bnd_K: lowest reasonable temp that */
                        /*      may correspond to theta_e at p_Pa   */
                        /* P_Pa: total pressure, Pa                  */
                        /* theta_e_knwn: Known equiv pot temp, K     */

    double w,                  /* mixing ratio, unitless; used for  */
                               /*  both actual and saturated        */
           theta_e_test,       /* Theta_e computed for t_it_K,      */
                               /*  compared to known theta_e        */
           t_it_K;             /* Iterative guess for temperature   */
    int    i;                  /* Iteration counter                 */
    double theta_e();
                               /* Starting at t_lwr_bnd_K, the temp */
                               /*  is increased by 0.5 degree       */
                               /*  increments until the solved-for  */
                               /*  theta_e first exceeds the known  */
                               /*  theta_e.  Therefore, the temp    */
                               /*  the known theta_e, P_Pa is found */
                               /*  to within 0.5 K                  */

    for(t_it_K = t_lwr_bnd_K,i=0;
        i<40 && theta_e_test < theta_e_knwn;
        t_it_K += 0.5, i++)
        {
        theta_e_test = theta_e(t_it_K,p_Pa);
        }
    if(i < 39)return(t_it_K);
    else return(-1.0);
    }

main()
{

printf("Input the data file name: ");   /* Where is the UA data?    */
scanf("%s",data_file_nm);

num_raw_lvls = input_ua(data_file_nm);
printf("Read %d levels",num_raw_lvls);

                               /* Designate the output file renamed */
                               /*  data_file_nm.ANL.  Output file   */
                               /*  is screen output saved to a file */
file_ext= data_file_nm;
len= strcspn(file_ext,".");
file_out = "        ";
strncpy(file_out,file_ext,len);
strcat(file_out,".ANL");
outfile= fopen(file_out,"w");
printf("  Output file: %s\n",file_out);
fprintf(outfile,"     PBE calculations from file %s:\n",data_file_nm);
```

```
for(i=0;i<num_raw_lvls;i++)    /* Perform unit conversions           */
   {
   t_raw_K[i] = C_to_K(t_raw_C[i]);
   if(td_raw_C[i] > -99.)      /* -99 is MM or drypoint              */
      {
      td_raw_K[i] = C_to_K(td_raw_C[i]);
      }
   else td_raw_K[i] = -999.;
   p_raw_Pa[i] = mb_to_Pa(p_raw_mb[i]*1.0);
   }
                               /* Section to interpolate raw data;  */
                               /*  Interpolation done linearly: by  */
                               /*  rules for obs and coding UA data */
                               /*  this is the only valid assumption*/

intrp_index = 0;               /* Start interpltd arrays at index 0 */
raw_data_index = 1;            /* So 1st interps will be between    */
                               /*  raw levels 0 & 1                 */
                               /* If data doesn't go to 100 mb, pick*/
                               /*  the last data level as the depth */
if(mean_depth_Pa < p_raw_Pa[num_raw_lvls-1])
   mean_depth_Pa= p_raw_Pa[num_raw_lvls-1];
else
   mean_depth_Pa= 10000.0;

for(p_intrp = p_raw_Pa[0]; p_intrp > mean_depth_Pa;
    p_intrp -= intrp_incr_Pa)
   {                           /* Raw data index is reset to the    */
                               /*  next lower input pressure as is  */
                               /*  needed to stay below the inter-  */
                               /*  polation pressure p_int_Pa       */
   if(p_intrp < p_raw_Pa[raw_data_index])raw_data_index++;
                               /* Find the interpolating fraction   */
   frac = (p_intrp - p_raw_Pa[raw_data_index-1]) /
          (p_raw_Pa[raw_data_index] - p_raw_Pa[raw_data_index-1]);
                               /* Interpolate temperature           */
   t_int_K[intrp_index] = t_raw_K[raw_data_index-1] + (frac *
          (t_raw_K[raw_data_index] - t_raw_K[raw_data_index-1]));
                               /* Interpolate mixing ratio          */
   if(td_raw_K[raw_data_index] != -999.)td_int_K =
          td_raw_K[raw_data_index-1] + (frac *
          (td_raw_K[raw_data_index] - td_raw_K[raw_data_index-1]));
   else td_int_K = -999.;

   if(td_int_K > 0)w_int[intrp_index] = mix_rat(td_int_K,p_intrp);
   else w_int[intrp_index] = 0.0;
                               /* Compute hydrostatic delta_z       */
   delta_z_m[intrp_index] = intrp_incr_Pa*287.05*t_int_K[intrp_index]/
          (p_intrp * g);
   p_int_Pa[intrp_index] = p_intrp;
```

```
                               /* Sum height, look for 0 C          */
   z_m += delta_z_m[intrp_index];

   if(t_int_K[intrp_index]<273.16 && t_int_K[intrp_index-1] >= 273.16)
     {
     fprintf(outfile,"\nMelting level: %5.0f m\n",z_m);
     printf("Melting level: %5.0f m\n",z_m);
     }
   intrp_index++;
   }

                               /* Compute low-level means, LCL, CCL */

sum_w = sum_theta = 0.0;

for(i = 0;
    p_int_Pa[i] > (p_int_Pa[0] - mean_depth_Pa) && i < intrp_index-1;
    i++)
   {
   sum_w += w_int[i];
   sum_theta += poisson_theta(t_int_K[i],p_int_Pa[i]);
   }
mean_w = sum_w / (i*1.0);
mean_theta = sum_theta / (i*1.0);
printf("\nMean conditions in lowest %5.1f mb:\n",mean_depth_Pa/100.);
printf("   mixing ratio: %4.1f g/kg\n",mean_w * 1000.);
printf("   potential temperature: %5.1f K\n",mean_theta);
fprintf(outfile,"\nMean conditions in lowest %5.1f mb:\n",
          mean_depth_Pa/100.);
fprintf(outfile,"   mixing ratio: %4.1f g/kg\n",mean_w * 1000.);
fprintf(outfile,"   potential temperature: %5.1f K\n",mean_theta);

                               /* Compute various important levels  */
                               /*  and some of the negative energy. */
                               /* Look from surface to 400 mb.      */

CCL_found = LCL_found = NO;
                               /* Check for data below 400 mb; if   */
                               /*  so, then assign the depth of the */
                               /*  low level layer to the last lvl  */
                               /*  in the sounding data; if not,    */
                               /*  then 400 mb is the depth of lyr. */
if(mean_depth_Pa > 40000.)
  {
  depth_low_level = mean_depth_Pa;
  printf("\n  Depth of low-level layer is below 400 mb\n");
  fprintf(outfile,"\n  Depth of low-level layer is below 400 mb\n");
  }
else
  depth_low_level = 40000.0;
```

```
for(i=0;p_int_Pa[i] > depth_low_level && CCL_found == NO;i++)
   {
                               /* If the low-level-mean mixing ratio*/
                               /*  is greater than the saturation   */
                               /*  mixing ratio on the temperature  */
                               /*  sounding, the mean mix ratio has */
                               /*  intersected the sounding, thus   */
                               /*  defining the CCL.                */
   if(mean_w >= mix_rat(t_int_K[i],p_int_Pa[i]) &&
      (CCL_found == NO || LCL_found == NO))
      {
      CCL_found = YES;
      p_CCL_Pa = p_int_Pa[i];  /* Save pressure                     */
                               /* Compute theta_CCL; this is the    */
                               /*  potential temp at CCL, at top    */
                               /*  of and thru depth of mixed layer */
      theta_CCL = poisson_theta(t_int_K[1],p_int_Pa[i]);
                               /* Compute convective temperature    */
      conv_temp_K = poisson_t(theta_CCL,p_int_Pa[0]);
      theta_e_CCL = theta_e(t_int_K[i],p_CCL_Pa);
      printf("\nCCL data -- pressure: %5.1f mb\n",p_CCL_Pa/100.);
      printf("   mixed-layer pot temp: %5.1f K\n",theta_CCL);
      printf("   convective temperature: %4.1f C\n",
               K_to_C(conv_temp_K));
      printf("   parcel equiv pot temp: %5.1f K\n",theta_e_CCL);
      fprintf(outfile,"\nCCL data -- pressure: %5.1f mb\n",
                  p_CCL_Pa/100.);
      fprintf(outfile,"   mixed-layer pot temp: %5.1f K\n",
                  theta_CCL);
      fprintf(outfile,"   convective temperature: %4.1f C\n",
                K_to_C(conv_temp_K));
      fprintf(outfile,"   parcel equiv pot temp: %5.1f K\n",
                   theta_e_CCL);
      }
                               /* Similarly, find the LCL by        */
                               /*  computing w at the temp repr by  */
                               /*  the low-level-mean pot temp, and */
                               /*  iterate until this w is less than*/
                               /*  the low-level-mean w             */
   if(LCL_found == NO)
      {
      t_on_mn_theta = poisson_t(mean_theta,p_int_Pa[i]);
      if(mean_w >= mix_rat(t_on_mn_theta,p_int_Pa[i]))
         {
         LCL_found = YES;
         p_LCL_Pa = p_int_Pa[i];
         theta_e_LCL = theta_e(t_on_mn_theta,p_LCL_Pa);
         printf("\nLCL data -- pressure: %5.1f mb\n",p_LCL_Pa/100.);
         printf("   parcel equiv pot temp: %5.1f K\n",theta_e_LCL);
         fprintf(outfile,"\nLCL data -- pressure: %5.1f mb\n",
              p_LCL_Pa/100.);
         fprintf(outfile,"   parcel equiv pot temp: %5.1f K\n",
                   theta_e_LCL);
         }
      }
   }
/*                      Compute P-CAPE                                  */
/*                                                                      */
/* Notes concerning computation and physical interpretation:           */
/* The "negative area" in this computation will be taken as the area*/
/* between the temp sounding and the dry adiabat at the potential     */
/* temperature of the mixed layer & CCL.  Therefore, thsi negative    */
/* area corresponds to the heat needed to form the mixed layer, and */
/* as such, deep enough to bring parcels to the LFC.  Often, most     */
/* of this area is removed by the addition of heat, and only a small*/
/* additional amount of energy is required from mechanical lifting  */
/* to bring the parcel to the LFC.                                     */
/* In this computation, the LFC is the CCL.  Above that level,        */
/* positive area (CAPE,PBE) is computed as the sum of                  */
/* [T(theta-e) - T]/T multiplied by gravity, where T(theta-e) is the*/
/* temperature on the "moiost adiabat" originating at the CCL.        */
/* T(theta-e) is solved iteratively: equivalent potential temp is    */
/* known, so T must be found to satisfy the equation                   */
/*  theta-e = theta * exp(Lw/CpT).                                     */

                               /* Compute negative area                */

sum_neg_area = 0.0;
for(i=0;i < 200 && p_int_Pa[i] > p_CCL_Pa;i++)
   {
   sum_neg_area += ((poisson_t(theta_CCL,p_int_Pa[i]) - t_int_K[i]) /
                    t_int_K[i]) * delta_z_m[i];
   }
sum_neg_area *= g;
printf("\nMixed-layer process data:\n   negative energy: %6.1f J/kg\n"
,
        sum_neg_area);
fprintf(outfile,"\nMixed-layer process data:\n");
fprintf(outfile,"   negative energy: %6.1f J/kg\n",sum_neg_area);

                               /* Compute CAPE.  Computation will      */
                               /*  end at first instance of negative*/
                               /*  buoyancy.                           */
cape = 0.0;
for(;i < intrp_index-1;i++)
   {
   parcel_t_K = t_of_theta_e(t_int_K[i],p_int_Pa[i],theta_e_CCL);
   if(parcel_t_K < 0.0)        /* Negative buoyancy                    */
      {
      printf("   equilibrium level: %5.1f mb\n",p_int_Pa[i]/100.);
      fprintf(outfile,"   equilibrium level: %5.1f mb\n",
               p_int_Pa[1]/100.);
      break;
      }
   pos_area = ((parcel_t_K - t_int_K[i])/ t_int_K[i]) * delta_z_m[i];
   cape += pos_area;
   }
cape *= g;
printf("   PBE: %6.1f J/kg\n",cape);
fprintf(outfile,"   PBE: %6.1f J/kg\n",cape);

/* The next section computes similar values (to the above), but uses*/
/* the assumption that parcels must be mechanically lifted to the    */
/* LFC, and that conditions prevailing at sounding time in the low-  */
/* levels are representative of those at the time of convection.     */
/* This is a good assumption if convection is occurring at around    */
/* the time of the sounding.  Negative area in this case is the area*/
/* between the temperature sounding and the low-level-mean adiabat   */
/* below the LCL.  Above the LCL, to the LFC, the negative area is   */
/* taken as the area between the moist adiabat above the LCL point   */
/* and the temperature sounding.  So this negative area corresponds */
/* to the integrated negative buoyancy experienced during forced     */
/* ascent to the LFC, or the mechanical energy that is needed to     */
/* lift the parcel to the LFC.                                       */
/*    Once at the LFC, the moist-adiabat is followed until the       */
/* parcel once again becomes negatively buoyant, as above ...        */

sum_neg_area = 0.0;
                               /* Negative area for the part from   */
                               /*  the surface to the LCL           */
for(i=0;i < 200 && p_int_Pa[i] > p_LCL_Pa;i++)
   {
   sum_neg_area += ((t_int_K[i] - poisson_t(mean_theta,p_int_Pa[i])) /
                    t_int_K[i]) * delta_z_m[i];
   }
                               /* And from the LCL to the LFC       */
parcel_t_K = poisson_t(mean_theta,p_LCL_Pa);
for(;i < 200;i++)
  {
                               /* Compute parcel temperature on     */
                               /*  moist adiabat above LCL, to LFC  */
  parcel_t_K = t_of_theta_e((parcel_t_K-1.0),p_int_Pa[i],theta_e_LCL);
  if(parcel_t_K > t_int_K[i]) break;
  sum_neg_area += ((t_int_K[i] - parcel_t_K) / t_int_K[i]) * delta_z_m
  [i];
  }
sum_neg_area *= g;
printf("\nLifting process data -- pressure at LFC: %5.1f mb\n",
         p_int_Pa[i]/100.);
printf("  negative energy: %6.1f J/kg\n",sum_neg_area);
fprintf(outfile,"\nLifting process data -- pressure at LFC: %5.1f mb\n
",
         p_int_Pa[i]/100.);
```

```
fprintf(outfile,"   negative energy: %6.1f J/kg\n",sum_neg_area);
                               /* Compute CAPE.  Computation will   */
                               /*  end at first instance of         */
                               /*  negaive buoyancy.                */
cape = 0.0;
for(;i < intrp_index-1;i++)
   {
   parcel_t_K = t_of_theta_e(t_int_K[i],p_int_Pa[i],theta_e_LCL);
   if(parcel_t_K < 0.0)        /* Negative buoyancy                 */
      {
      printf("   equilibrium level:  %5.1f mb\n",p_int_Pa[i]/100.);
      fprintf(outfile,"   equilibrium level:  %5.1f mb\n",
              p_int_Pa[i]/100.);
      break;
      }
   pos_area= ((parcel_t_K - t_int_K[i]) / t_int_K[i]) * delta_z_m[i];
   cape += pos_area;
   }
cape *= g;
printf("   PBE: %6.1f J/kg\n",cape);
fprintf(outfile,"   PBE: %6.1f J/kg\n",cape);
}
```

Appendix 6.5

UMAX, MDA, PBE, Shear Results For All Nine Outbreaks

This appendix lists the predictor variables used in the regression schemes described in Chapter 2. Note analyses times are in Z, UMAX is m s^{-1}, MDA is in degrees, low-level shear is s^{-1} and PBE is m^2 s^{-2}. Raob stations denoted with an asterisk (*) indicate 00Z soundings, otherwise 12Z soundings were used in the analysis. Cells annotated with a plus sign (+) indicate tornadic cells.

Table 6.6.1. Oklahoma Outbreak of 26 April 1984 (84117)

Cell	Time	Raob Station	UMAX	MDA	Shear	PBE
A	1930-2030	72456	10	-4	0.01258	1514.1
B	1930-2030	72553	15	15	0.01019	3506.2
C+	2014-2044	72553	20	15	0.01019	3506.2
D	2014-2044	72553	10	11	0.01019	3506.2
E	2014-2044	72456	13	14	0.01258	1514.1
F	2044-2130	72456	18	11	0.01258	1514.1
G	2044-2130	72353	8	-5	0.01379	1399.0
H	2200-2230	72353	5	5	0.01379	1399.0
I	2200-2300	72353	4	-4	0.01379	1399.0
J+	2200-2300	72353	18	13	0.01379	1399.0
K	2200-2300	72456,72553	12	17	0.01138	2510.2
L	2200-2300	72451,72353	5	7	0.01421	881.6
M	2230-2300	72456	8	7	0.01258	1514.1
N	2230-2300	72456	13	0	0.01258	1514.1
O	2230-2300	72456	10	5	0.01258	1514.1
P+	2053-2130	72562	15	21	0.00939	397.0

Table 6.6.2. Wisconsin-Illinois Outbreak of 27 April 1984 (84118)

Cell	Time	Raob Station	UMAX	MDA	Shear	PBE
A	1715-1800	72433	15	16	0.00782	2070.2
B	2000-2100	72433	12	16	0.00782	2070.2
C	2030-2100	72532	10	15	0.00679	4121.4
D	2100-2200	72433	13	10	0.00782	2070.2
D'+	2230-2330	72433	15	30	0.00782	2070.2
E	2130-2230	72532	16	22	0.00679	4121.4
E'+	2230-2330	72532	33	32	0.00679	4121.4
F	2130-2230	72532,72645	30	7	0.00926	2221.2
F'+	2230-2300	72532,72645	27	35	0.00926	2221.2
G	2130-2200	72645	20	16	0.01478	321.0
H	2145-2245	72532	15	22	0.00679	4121.4
J+	2230-2300	72532	16	34	0.00679	4121.4
N+	2100-2200	72645	30	51	0.01478	321.0
Q	2100-2130	72349	0	17	0.00729	1787.5
S	2215-2245	72349	8	10	0.00729	1787.5
T	1600-1700	72349	12	21	0.00729	1787.5

Table 6.6.3. Iowa Outbreak of 7 June 1984 (84159)

Cell	Time	Raob Station	UMAX	MDA	Shear	PBE
A+	1931-2030	72456,72553	22	11	0.00619	1794.1
B+	2000-2100	72456	25	13	0.00745	1071.0
C+	2000-2100	72456	27	15	0.00745	1071.0
D+	1901-2000	72553	50	15	0.00494	2517.1
E	2200-2230	72456,72553	12	8	0.00619	1794.1
E'	2230-2330	72456,72553	12	0	0.00619	1794.1
G	1801-1901	72654	12	5	0.00547	1822.8
H	1801-1901	72654	12	8	0.00547	1822.8
I	1701-1801	72562,72654	15	-1	0.00511	1771.3
J+	2245-2330	72456	24	13	0.00745	1071.0
J'+	2245-2330	72456	20	12	0.00745	1071.0
L	1901-2000	72553	8	5	0.00494	2517.1

Table 6.6.4. Nebraska Outbreak of 10 May 1985 (85130)

Cell	Time	Raob Station	UMAX	MDA	Shear	PBE
A+	2000-2100	72451	15	21	0.00866	2129.5
B	2000-2100	72451,72562	7	-9	0.00705	2073.6
C	2000-2100	72456	5	-3	0.00418	1400.1
D	2100-2200	72562	5	13	0.00545	2017.6
E	2130-2230	72562	10	-13	0.00545	2017.6
F+	2144-2244	72562,72451	20	12	0.00705	2073.6
G+	2200-2300	72451	30	18	0.00866	2129.5
H	2130-2230	72451	10	-3	0.00866	2129.5
I	2200-2300	72469,72562	2	-2	0.00549	1348.8
J	2230-2330	72469	2	10	0.00534	680.0
K	2314-2344	72469	4	-1	0.00534	680.0
L	2214-2314	72562	8	2	0.00545	2017.6
M+	2314-2344	72562	20	9	0.00545	2017.6

Table 6.6.5. Ohio-Pennsylvania-New York Outbreak of 31 May 1985 (85150)

Cell	Time	Raob Station	UMAX	MDA	Shear	PBE
A	2000-2100	72520,72528	15	17	0.00782	628.3
B(C)+	2000-2100	72520,72528	38	38	0.00782	628.3
D	2000-2100	72528	2	9	0.00521	777.3
E	2030-2100	72520	10	-5	0.01043	479.3
F	2030-2100	72429,72520	3	2	0.00968	903.6
G+	2130-2230	72429,72520	25	36	0.00968	903.6
H+	2130-2230	72429,72520	25	54	0.00968	903.6
I+	2200-2300	72429,72520	26	29	0.00968	903.6
J2+	2300-2344	72429	31	16	0.00892	1327.8
K+	2230-2330	72429	20	25	0.00892	1327.8
L	2230-2330	72429	10	14	0.00892	1327.8
M+	2300-2344	72429	20	29	0.00892	1327.8
O+	2300-2344	72520	20	28	0.01043	479.3
Q	2030-2100	72520	5	9	0.01043	479.3
R	2300-2344	72429,72327	6	15	0.00716	1650.3
U	2030-2100	72429	10	-7	0.00892	1327.8
V	2000-2100	72425	15	18	0.00544	1409.6
W	2030-2130	72425,72327	19	15	0.00542	1691.2
X	2130-2230	72425,72520	10	22	0.00794	944.5
Y	2130-2230	72425	2	18	0.00544	1409.6
Z	2130-2230	72327	15	25	0.00540	1972.8
AA	2144-2244	72327	22	11	0.00540	1972.8

Table 6.6.6. South Dakota Outbreak of 28 July 1986 (86209)

Cell	Time	Raob Station	UMAX	MDA	Shear	PBE
A	1714-1800	72764,72654	8	-8	0.00657	2461.8
B	1830-1900	72655	10	5	0.00714	320.2
C	1900-2000	72655	13	0	0.00714	320.2
D	1900-2000	72655	9	6	0.00714	320.2
E	1914-2000	72655	13	17	0.00714	320.2
F	1914-2000	72662,72764	5	12	0.00699	1671.6
G	1930-2000	72662,72764	12	8	0.00699	1671.6
H+	1930-2000	72654	20	23	0.00494	3046.2
I+	2000-2100	72654	10	24	0.00494	3046.2
J	2130-2230	72662,72764	10	-6	0.00699	1671.6
K	2130-2230	72662,72764	10	-8	0.00699	1671.6
L+	2200-2230	72553,72654	18	23	0.00467	2845.3

Table 6.6.7. Kansas Outbreak of 18 September 1986 (86261)

Cell	Time	Raob Station	UMAX	MDA	Shear	PBE
A	2001-2101	72363	7	13	0.00696	2669.9
B	2001-2101	72363	7	-3	0.00696	2669.9
C	2001-2101	72662	15	-6	0.00829	202.0
D+	2101-2201	72469,72662	23	15	0.00686	255.1
E	2131-2231	72451	8	18	0.00310	2212.3
G	2131-2231	72451	5	13	0.00310	2212.3
H	2131-2231	72562,72654	20	-1	0.00962	975.5
I	2131-2231	72469	15	-3	0.00543	308.2
J	2201-2301	72469	8	2	0.00543	308.2
K	2231-2301	72553	8	-7	0.00739	654.4
L	0001-0101	72451	15	-8	0.00310	2212.3
M+	0001-0101	72456	17	10	0.00818	4770.4
N	0001-0101	72562,72451	13	1	0.00587	1261.2
O+	0001-0101	72469	15	24	0.00543	308.2
P	0001-0101	72456,72553	10	-4	0.00818	2712.4
Q+	0101-0201	72456	16	11	0.00818	4770.4

Table 6.6.8. Texas-Louisiana Outbreak
of 15 November 1987 (87319)

Cell	Time	Raob Station	UMAX	MDA	Shear	PBE
A+	1531-1631	72255	20	16	0.00894	2253.0
B+	1531-1631	72255	15	17	0.00894	2253.0
C	1601-1701	72255	13	10	0.00894	2253.0
D	1601-1701	72260	2	18	0.00905	390.9
E	1631-1731	72255	15	11	0.00894	2253.0
F	1916-2016	72255,72260	10	18	0.00900	1322.0
G	1831-1901	72260	8	21	0.00905	390.9
H+	2201-2301	72255	30	21	0.00894	2253.0
I	2101-2201	72255	15	4	0.00894	2253.0
J+	2231-2331	72240	18	14	0.00653	535.7
K	2201-2231	72255	13	9	0.00894	2253.0
L	2046-2201	72261,72260	17	13	0.00743	1028.9
M	1946-2016	72255,72240	13	1	0.00774	1394.4
N	2201-2301	72255,72240	10	7	0.00774	1394.4
O	2316-2346	72255,72240	15	14	0.00774	1394.4
P	0031-0131	72255	13	7	0.00894	2253.0

Table 6.6.9. North Carolina-Virginia Outbreak
of 28 November 1988 (88333)

Cell	Time	Raob Station	UMAX	MDA	Shear	PBE
A+	0501-0601	*72208,72311	10	42	0.01170	1335.3
B+	0731-0831	*72208,72311	15	13	0.01170	1335.3
C	0001-0101	*72317	5	11	0.01068	321.2
D	0001-0101	*72317	0	-1	0.01068	321.2
E	0001-0101	*72311	5	-5	0.01465	945.6
F	0031-0131	*72402	0	13	0.00991	321.9
G	0131-0231	*72317	5	-6	0.01068	321.2
H+	0431-0531	*72317	6	15	0.01068	321.2
I	0601-0701	*72317	10	-4	0.01068	321.2
J	0531-0631	*72402	0	2	0.00991	321.9
K	0531-0631	*72208,72311	10	4	0.01170	1335.3
L	0631-0731	*72208,72311	10	3	0.01170	1335.3
M	0501-0601	*72213	3	6	0.00836	1549.0

Appendix 6.6

Explanation of Statistical Procedures and Tests

This appendix defines the statistical tests used in the procedure determining whether outbreak intercepts were the same in a given data set (Chapter 3). An explanation of the results is also given.

By using the statistical package SAS (SAS, 1985), a regression model was chosen relating only tornadic occurrence to the predictor variables UMAX and MDA. The classification of tornadic and non-tornadic occurrence was denoted as a "0" or a "1". If the observed cell in the data set had a positive value for F_w then that cell was classified as tornadic, or "1". If the cell had a value of 0 for F_w then the cell was classified as non-tornadic, or "0". For the purposes of the model used, this variable was denoted as GROUP. As a result, the variables UMAX and MDA describe how a constant slope fits the data.

In addition to the variables UMAX and MDA, another variable was put into the model, denoted as OUTBREAK, to classify outbreaks within the data set. A numbering system was used to designate one outbreak from another. For example, in the combined data set, the variable OUTBREAK had a value range of 1 through 9 corresponding to the particular outbreak in the data. This OUTBREAK variable then explains how the outbreak intercepts relate to one another. The interaction of UMAX and MDA upon OUTBREAK was also tested. This interaction basically describes how varying slopes of UMAX and MDA fit the data. In SAS notation, this model was run under the GLM procedure and was set up as the following:

```
PROC GLM;
   MODEL GROUP = UMAX MDA OUTBREAK UMAX*OUTBREAK MDA*OUTBREAK;
```

By running this model on SAS, a number of test statistics result describing the statistical relationship of UMAX, MDA and OUTBREAK to tornado occurrence. One such test statistic is the p-value. Simply stated, it is the probability that the model test statistic would be greater than the actual computed test statistic (under the hypothesis that UMAX and MDA are not needed in the model). As a result, a small p-value would tend to support the analysis that the variables are not the same and would therefore be needed in the data. Conversely, a large p-value would mean that the variables are not needed in the analysis, and have little significance as to the outcome of the test.

Another way to determine the significance of a test is by the use of the Type I sums of squares, or sequential sums of squares. They are the incremental improvements in error sums of squares as each effect (or variable) is added to the model (SAS, 1985). If the Type I sums of squares is relatively high, then that effect would be kept in the data. If it was relatively low, then that effect could be left out of the model. For example, if three effects, or variables, are used in the model (UMAX, MDA and OUTBREAK) then the Type I sums of squares explains the significance of the desired effect as each effect is added to the model. An analysis output example is given in table 6.6.1.

This analysis shows that there is a difference in UMAX and MDA from outbreak to outbreak (hence the low p-value and the relatively high Type I sums of squares). Also, the variable OUTBREAK shows a strong difference from outbreak to outbreak, suggesting that each outbreak has a different intercept, and consequently, a different surface. However,

Table 6.6.1. Example of test of hypothesis output using SAS with the appropriate p-value and Type I sums of squares for the effect shown.

Effect	degrees of freedom	Type I SS	p-value
UMAX	1	11.9258	0.0001
MDA	1	3.0655	0.0001
OUTBREAK	8	3.0132	0.0001
UMAX*OUTB	8	0.7257	0.3470
MDA*OUTB	8	0.7801	0.2952

when the effects of UMAX and MDA upon OUTBREAK are added to the model, the differences in those relationships to the data appears insignificant.

The above analysis tells us that as the values of UMAX and MDA increase, so does the likelihood of a tornadic occurrence. This falls in line with the basic idea of the breakpoint value. That is, the higher the value of UMAX and MDA, the more likely that observation is going to fall on the rising-ridge surface describing tornadic intensity. Similarly, outbreaks are significantly different from one another, suggesting different surfaces are going to relate tornadic occurrence differently, and are therefore needed in the analysis.

Appendix 6.7

Statistical Results Using F_w^2

Table 6.7.1. Percent of variation (coefficient of determination or R^2) using F_w^2 that can be explained by the predictor variables UMAX, MDA, shear and PBE for linear bivariate and quadvariate regression.

		F_w^2
CASE	PREDICTOR	LINEAR
OK	UMAX,MDA,SHEAR,PBE	0.4284
84117	SHEAR,PBE	0.2854
	UMAX,MDA	0.3680
WI-ILL	UMAX,MDA,SHEAR,PBE	0.8520
84118	SHEAR,PBE	0.3276
	UMAX,MDA	0.8031
IA	UMAX,MDA,SHEAR,PBE	0.8059
84159	SHEAR,PBE	0.5197
	UMAX,MDA	0.6412
NE	UMAX,MDA,SHEAR,PBE	0.8774
85130	SHEAR,PBE	0.4810
	UMAX,MDA	0.8270
OH-PA	UMAX,MDA,SHEAR,PBE	0.8009
85151	SHEAR,PBE	0.1369
	UMAX,MDA	0.7267
SD	UMAX,MDA,SHEAR,PBE	0.9815
86209	SHEAR,PBE	0.9649
	UMAX,MDA	0.7268
KS	UMAX,MDA,SHEAR,PBE	0.7253
86261	SHEAR,PBE	0.2178
	UMAX,MDA	0.5531
TX-LA	UMAX,MDA,SHEAR,PBE	0.8306
87319	SHEAR,PBE	0.0614
	UMAX,MDA	0.7490
NC	UMAX,MDA,SHEAR,PBE	0.8525
88333	SHEAR,PBE	0.1003
	UMAX,MDA	0.8254

		F_w^2	
CASE	PREDICTOR	LINEAR	QUAD.
COMBINED	UMAX,MDA,SHEAR,PBE	0.5589	0.6614
DATA	SHEAR,PBE	0.0125	0.0519
	UMAX,MDA	0.5511	0.6255
COMBINED	UMAX,MDA,SHEAR,PBE	0.5818	0.7216
NON-DIM.	UMAX,MDA	0.5726	0.6705
STRATIFIED	UMAX,MDA,SHEAR,PBE	0.7813	0.9560
CASE I	SHEAR,PBE	0.1974	0.2320
(WI;OH)	UMAX,MDA	0.7107	0.9134
STRATIFIED	UMAX,MDA,SHEAR,PBE	0.5976	0.8920
CASE II	SHEAR,PBE	0.2157	0.4038
(OK;IA)	UMAX,MDA	0.5132	0.7508
STRATIFIED	UMAX,MDA,SHEAR,PBE	0.6838	0.9697
CASE III	SHEAR,PBE	0.0906	0.1750
(SD;TX)	UMAX,MDA	0.6502	0.8821
STRATIFIED	UMAX,MDA,SHEAR,PBE	0.7230	0.9125
CASE IV	SHEAR,PBE	0.1739	0.3054
(KS;NE;NC)	UMAX,MDA	0.6740	0.8062

REFERENCES

Anderson, C. E., 1979: Anvil outflow patterns as indicators of tornadic thunderstorms. Preprints 11th Conference on Severe Local Storms, Oct 2-5, 1979 (Kansas City, MO), AMS, Boston, MA, 481-485.

____, and K. J. Schrab, 1988: Use of satellite imagery to identify· tornadic thunderstorms. Preprints 15th Conference on Severe Local Storms, Feb 22-26, 1988 (Baltimore, MD), AMS, Boston, MA, 186-189.

Blechman, J. B., 1979: Vortex generation in a nested grid numerical thunderstorm model. Ph. D. Thesis. University of Wisconsin-Madison.

Box, G. E. P. and N. R. Draper, 1987: Emperical Model-Building and Response Surfaces. J. Wiley & Sons, New York, 669 pp.

Colquhoun, J. R. and D. J. Shepherd, 1985: The relationship between tornado intensity and the environment of its parent severe thunderstorm. Preprints 14th Conference on Severe Local Storms, Oct 29-Nov 1, 1985 (Indianapolis, IN), AMS, Boston, MA, 1-4.

Davies-Jones, R. P., 1982: Tornado dynamics, Thunderstorms: A Social, Scientific, & Technological Documentary -- Thunderstorm Morphology and Dynamics, Vol. 2., February 1982, NOAA/ERL, Washington, D.C., 297-363.

____, 1983: The onset of rotation in thunderstorms. Preprints 13th Conference on Severe Local Storms, Oct 17-20, 1983 (Tulsa, OK), AMS, Boston, MA, 215-218.

Department of the Air Force, 1969: Use of the Skew-T, Log P diagram in analysis and forecasting. AWS Manual 105-124, 15 July 1969, Headquarters, Air Weather Service (MAC), Scott AFB, IL 62225, 138 pp.

Fujita, T. F., 1981: Tornadoes and downbursts in the context of generalized planetary scales. J. Atmos. Sci., **38**, 1511-1534.

____, 1988: Private communication.

Galway, J. G., 1975: Relationship of tornado deaths to severe watch areas. Mon. Wea. Rev., **103**, 737-741.

____, 1977: Some climatological aspects of tornado outbreaks. Mon. Wea. Rev., **105**, 477-484.

____, 1983: Tornado track characteristics of killer storms. Preprints 13th Conference on Severe Local Storms, Oct 17-20, 1983 (Tulsa, OK), AMS, Boston, MA, 117-119.

Glossary of Meteorology, 1959. Edited by R. E. Huschke. AMS, Boston, MA, 638 pp.

Leftwich, P. W. Jr., and X. Wu, 1988: An operational index of the potential for violent tornado development. Preprints 15th Conference on Severe Local Storms, Feb 22-26, 1988 (Baltimore, MD), AMS, Boston, MA, 472-475.

Maddox, R. A., and C. A. Doswell III, 1982: Forecasting severe thunderstorms: a brief evaluation of accepted techniques. Preprints 12th Conference on Severe Local Storms, Jan 12-15, 1982 (San Antonio, TX), AMS, Boston, MA, 92-95.

Miller, R. C., 1972: Notes of analysis and severe storms forecasting procedures of the Air Force Global Weather Central. AWS Tech. Report 200 (Rev.), Headquarters Air Weather Service, Scott AFB, 94 pp.

National Climatic Data Center, 1984: Storm Data, **26**, 4, 1-61.

National Climatic Data Center, 1984: Storm Data, **26**, 6, 1-59.

National Climatic Data Center, 1985: Storm Data, **27**, 5, 1-51.

National Climatic Data Center, 1986: Storm Data, **28**, 7, 1-65.

National Climatic Data Center, 1986: Storm Data, **28**, 10, 1-63.

National Climatic Data Center, 1987: Storm Data, **29**, 11, 1-51.

Pautz, M. E., 1969: Severe local storm occurrences, 1955-1967. ESSA Tech. Memo. WBTM FCST12, Washington, D.C., 3-4.

Rasmussen, E. N., and R. B. Wilhelmson, 1983: Relationships between storm characteristics and 1200 GMT hodographs, low-level shear, and stability. Preprints 13th Conference on Severe Local Storms, Oct 17-20, 1983 (Tulsa, OK), AMS, Boston, MA, J5-J8.

____, 1989: Private communication.

SAS Institute, 1985: SAS User's Guide: Statistics, Version 5 Edition, SAS Institute Inc., Cary, NC, 956 pp.

Schrab, K. J., 1988: A study on the use of satellite imagery to identify tornadic intensity of thunderstorms. M. S. Thesis. University of Wisconsin-Madison.

Weisman, M. J., and J. B. Klemp, 1982: The dependence of numerically simulated convective storms on vertical wind shear and buoyancy. Mon. Wea. Rev., **110**, 504-520.